

### Схожість

Веб джерела 91

1	<b>uk.wikijaa.ru</b> <a href="https://uk.wikijaa.ru/wiki/Website">https://uk.wikijaa.ru/wiki/Website</a>	6.4%
2	<b>ewikiuk.top</b> <a href="https://ewikiuk.top/wiki/Website">https://ewikiuk.top/wiki/Website</a>	6.4%
3	<b>uk.wikijaa.ru</b> <a href="https://uk.wikijaa.ru/wiki/Class_diagram">https://uk.wikijaa.ru/wiki/Class_diagram</a>	4.87%
4	<b>ewikiuk.top</b> <a href="https://ewikiuk.top/wiki/Class_diagram">https://ewikiuk.top/wiki/Class_diagram</a>	4.87%
5	<b>uk.wikijaa.ru</b> <a href="https://uk.wikijaa.ru/wiki/Web_development">https://uk.wikijaa.ru/wiki/Web_development</a>	2.9%
6	<b>uk.wikijaa.ru</b> <a href="https://uk.wikijaa.ru/wiki/Java_(programming_language)">https://uk.wikijaa.ru/wiki/Java_(programming_language)</a>	1.3%
7	<b>ewikiuk.top</b> <a href="https://ewikiuk.top/wiki/Java_(programming_language)">https://ewikiuk.top/wiki/Java_(programming_language)</a>	1.3%
8	<b>er.knutd.edu.ua</b> <a href="https://er.knutd.edu.ua/bitstream/123456789/18045/1/Diplom123_Kharenko_Yarmolenko.pdf">https://er.knutd.edu.ua/bitstream/123456789/18045/1/Diplom123_Kharenko_Yarmolenko.pdf</a>	0.89%
9	<b>uk.living-in-belgium.com</b> <a href="https://uk.living-in-belgium.com/webpage-vs-website-10569">https://uk.living-in-belgium.com/webpage-vs-website-10569</a>	0.82%
10	<b>stackoverflow.com</b> <a href="https://stackoverflow.com/questions/70713657/cannot-override-bootstrap-theme">https://stackoverflow.com/questions/70713657/cannot-override-bootstrap-theme</a>	0.77%

# Диплом.СельвесюкІ.І.

Завантажено: 06/19/2022 | Перевірено: 06/19/2022

● Схожість ● Цитата ● Використані джерела ● Заміна символів

11	<b>uk.wikijaa.ru</b> <a href="https://uk.wikijaa.ru/wiki/Jakarta_Servlet">https://uk.wikijaa.ru/wiki/Jakarta_Servlet</a>	0.61%
12	<b>eprints.library.odeku.edu.ua</b> <a href="http://eprints.library.odeku.edu.ua/id/eprint/3774/1/Kopelchuk_sozdanie_internet_magazina_M_2019.pdf">http://eprints.library.odeku.edu.ua/id/eprint/3774/1/Kopelchuk_sozdanie_internet_magazina_M_2019.pdf</a>	0.55%
13	<b>wikizero.com</b> <a href="https://wikizero.com/uk/HTML">https://wikizero.com/uk/HTML</a>	0.49%
14	<b>uk.wikipedia.org</b> <a href="https://uk.wikipedia.org/wiki/HTML">https://uk.wikipedia.org/wiki/HTML</a>	0.49%
15	<b>uk.wikijaa.ru</b> <a href="https://uk.wikijaa.ru/wiki/HTML_attribute">https://uk.wikijaa.ru/wiki/HTML_attribute</a>	0.45%
16	<b>ewikiuk.top</b> <a href="https://ewikiuk.top/wiki/Jakarta_Server_Pages">https://ewikiuk.top/wiki/Jakarta_Server_Pages</a>	0.42%
17	<b>uk.wikijaa.ru</b> <a href="https://uk.wikijaa.ru/wiki/Jakarta_Server_Pages">https://uk.wikijaa.ru/wiki/Jakarta_Server_Pages</a>	0.42%
18	<b>ikt.in.ua</b> <a href="https://ikt.in.ua/category/bloh-vchytelya-informatyky-svidryk-vasylya-evhenovycha/8-klas/">https://ikt.in.ua/category/bloh-vchytelya-informatyky-svidryk-vasylya-evhenovycha/8-klas/</a>	0.38%
19	<b>ikt.in.ua</b> <a href="https://ikt.in.ua/category/bloh-vchytelya-informatyky-svidryk-vasylya-evhenovycha/">https://ikt.in.ua/category/bloh-vchytelya-informatyky-svidryk-vasylya-evhenovycha/</a>	0.38%
20	<b>pubhtml5.com</b> <a href="https://pubhtml5.com/zfej/khsd/basic">https://pubhtml5.com/zfej/khsd/basic</a>	0.25%
21	<b>en.wikipedia.org</b> <a href="https://en.wikipedia.org/wiki/HTML">https://en.wikipedia.org/wiki/HTML</a>	0.25%
22	<b>en-academic.com</b> <a href="https://en-academic.com/dic.nsf/enwiki/7889">https://en-academic.com/dic.nsf/enwiki/7889</a>	0.25%
23	<b>webarchive.library.unt.edu</b> <a href="https://webarchive.library.unt.edu/web/20170105064255/https://en.wikipedia.org/wiki/HTML">https://webarchive.library.unt.edu/web/20170105064255/https://en.wikipedia.org/wiki/HTML</a>	0.25%
24	<b>uk.wikijaa.ru</b> <a href="https://uk.wikijaa.ru/wiki/Semantic_HTML">https://uk.wikijaa.ru/wiki/Semantic_HTML</a>	0.24%
25	<b>ekmair.ukma.edu.ua</b> <a href="http://ekmair.ukma.edu.ua/bitstream/handle/123456789/18728/Stakhurskyi_Bakalavska_robota.pdf?sequence=1">http://ekmair.ukma.edu.ua/bitstream/handle/123456789/18728/Stakhurskyi_Bakalavska_robota.pdf?sequence=1</a>	0.21%
26	<b>host-base.ru</b> <a href="https://host-base.ru/linux/ustanovka-mysql-v-debian-10.html">https://host-base.ru/linux/ustanovka-mysql-v-debian-10.html</a>	0.2%
27	<b>www.hmong.press</b> <a href="https://www.hmong.press/wiki/MYSQL">https://www.hmong.press/wiki/MYSQL</a>	0.2%
28	<b>dev.business-russia-eeen.ru</b> <a href="http://dev.business-russia-eeen.ru/en/success-stories/mysql-111/">http://dev.business-russia-eeen.ru/en/success-stories/mysql-111/</a>	0.15%
29	<b>WikiMili.com</b> <a href="https://WikiMili.com/en/HTML">https://WikiMili.com/en/HTML</a>	0.13%

# Диплом.СельвесюкІ.І.

Завантажено: 06/19/2022 | Перевірено: 06/19/2022

● Схожість ● Цитата ● Використані джерела ● Заміна символів

30	<b>mymemory.translated.net</b> <a href="https://mymemory.translated.net/en/English/Russian/ecomstation">https://mymemory.translated.net/en/English/Russian/ecomstation</a>	0.12%
31	<b>docshare.tips</b> <a href="http://docshare.tips/computer-networking-principles-protocols-and-practice_575ab5c2b6d87fab218b4796.html">http://docshare.tips/computer-networking-principles-protocols-and-practice_575ab5c2b6d87fab218b4796.html</a>	0.12%
32	<b>translate.yandex.ru</b> <a href="https://translate.yandex.ru/translate?lang=en-ru&amp;url=https%3A%2F%2Fstackoverflow.com%2Fquestions%2F39399377%2F...">https://translate.yandex.ru/translate?lang=en-ru&amp;url=https%3A%2F%2Fstackoverflow.com%2Fquestions%2F39399377%2F...</a>	0.1%
33	<b>slidetodoc.com</b> <a href="https://slidetodoc.com/cascading-style-sheet-css-1-cascading-style-sheets-2/">https://slidetodoc.com/cascading-style-sheet-css-1-cascading-style-sheets-2/</a>	0.1%
34	<b>ep3.nuwm.edu.ua</b> <a href="http://ep3.nuwm.edu.ua/14902/1/05-02-279.pdf">http://ep3.nuwm.edu.ua/14902/1/05-02-279.pdf</a>	0.09%
35	<b>org2.knuba.edu.ua</b> <a href="https://org2.knuba.edu.ua/pluginfile.php/56881/mod_resource/content/1/%D0%9D%D0%B0%D0%B2%D1%87%D0%B0%D...">https://org2.knuba.edu.ua/pluginfile.php/56881/mod_resource/content/1/%D0%9D%D0%B0%D0%B2%D1%87%D0%B0%D...</a>	0.09%
36	<b>revolution.allbest.ru</b> <a href="https://revolution.allbest.ru/law/00627581_0.html">https://revolution.allbest.ru/law/00627581_0.html</a>	0.09%
37	<b>gendocs.ru</b> <a href="https://gendocs.ru/v32732/%D0%B0%D0%B2%D1%82%D0%BE%D0%BC%D0%B0%D1%82%D0%B8%D0%B7%D0%B0%...">https://gendocs.ru/v32732/%D0%B0%D0%B2%D1%82%D0%BE%D0%BC%D0%B0%D1%82%D0%B8%D0%B7%D0%B0%...</a>	0.07%
38	<b>drainages.narod.ru</b> <a href="http://drainages.narod.ru/olderfiles/1/WB_posibnik.pdf">http://drainages.narod.ru/olderfiles/1/WB_posibnik.pdf</a>	0.07%
39	<b>alversch.ru</b> <a href="http://alversch.ru/proizvodstvo-i-texnologii/vigotvleniya-virobiv-littyam.html">http://alversch.ru/proizvodstvo-i-texnologii/vigotvleniya-virobiv-littyam.html</a>	0.07%
40	<b>antibotan.com</b> <a href="https://antibotan.com/file.html?work_id=521750">https://antibotan.com/file.html?work_id=521750</a>	0.07%
41	<b>diplomba.ru</b> <a href="https://diplomba.ru/work/1404">https://diplomba.ru/work/1404</a>	0.07%
42	<b>www.100balov.com</b> <a href="http://www.100balov.com/data23/ukr/IJno_dlya_styidenta__14/1_2316.php">http://www.100balov.com/data23/ukr/IJno_dlya_styidenta__14/1_2316.php</a>	0.07%
43	<b>antibotan.com</b> <a href="https://antibotan.com/file.html?work_id=520564">https://antibotan.com/file.html?work_id=520564</a>	0.07%
44	<b>antibotan.com</b> <a href="https://antibotan.com/file.html?work_id=517771">https://antibotan.com/file.html?work_id=517771</a>	0.07%
45	<b>antibotan.com</b> <a href="https://antibotan.com/file.html?work_id=517770">https://antibotan.com/file.html?work_id=517770</a>	0.07%
46	<b>xn--r1a3b.xn--b1amgblet.xn--j1amh</b> <a href="http://xn--r1a3b.xn--b1amgblet.xn--j1amh/index.php/HTML">http://xn--r1a3b.xn--b1amgblet.xn--j1amh/index.php/HTML</a>	0.07%
47	<b>infopedia.su</b> <a href="https://infopedia.su/11x9499.html">https://infopedia.su/11x9499.html</a>	0.07%
48	<b>www.100balov.com</b> <a href="http://www.100balov.com/data23/ukr/Materialij_po_navchannu_pakynok_12/Referat_venchymijii_biznes.php">http://www.100balov.com/data23/ukr/Materialij_po_navchannu_pakynok_12/Referat_venchymijii_biznes.php</a>	0.07%

# Диплом.СельвесюкІ.І.

Завантажено: 06/19/2022 | Перевірено: 06/19/2022

● Схожість ● Цитата ● Використані джерела ● Заміна символів

49	<b>ep3.nuwm.edu.ua</b> <a href="http://ep3.nuwm.edu.ua/18240/1/04-03-281.pdf">http://ep3.nuwm.edu.ua/18240/1/04-03-281.pdf</a>	0.07%
50	<b>refeteka.ru</b> <a href="http://refeteka.ru/r-169245.html">http://refeteka.ru/r-169245.html</a>	0.07%
51	<b>www.management.org.ua</b> <a href="http://www.management.org.ua/wp-content/uploads/2017/02/prog_pract_new_edition.pdf">http://www.management.org.ua/wp-content/uploads/2017/02/prog_pract_new_edition.pdf</a>	0.07%
52	<b>nashaucheba.ru</b> <a href="https://nashaucheba.ru/v45206/%D0%BC%D0%B0%D1%80%D1%82%D0%B8%D0%BD%D1%8E%D0%BA_%D0%BF.%D0...">https://nashaucheba.ru/v45206/%D0%BC%D0%B0%D1%80%D1%82%D0%B8%D0%BD%D1%8E%D0%BA_%D0%BF.%D0...</a>	0.07%
53	<b>leksii.org</b> <a href="https://leksii.org/15-15176.html">https://leksii.org/15-15176.html</a>	0.07%
54	<b>otherreferats.allbest.ru</b> <a href="https://otherreferats.allbest.ru/miscellaneous/00006544_0.html">https://otherreferats.allbest.ru/miscellaneous/00006544_0.html</a>	0.07%
55	<b>uchika.in.ua</b> <a href="https://uchika.in.ua/motivaciya-personalu.html">https://uchika.in.ua/motivaciya-personalu.html</a>	0.07%
56	<b>ua-referat.com</b> <a href="https://ua-referat.com/%D0%9A%D0%BE%D0%BD%D1%82%D1%80%D0%BE%D0%BB%D1%8C_%D0%B4%D0%BE%D1...">https://ua-referat.com/%D0%9A%D0%BE%D0%BD%D1%82%D1%80%D0%BE%D0%BB%D1%8C_%D0%B4%D0%BE%D1...</a>	0.07%
57	<b>techical-documentation-page.netlify.app</b> <a href="https://techical-documentation-page.netlify.app/">https://techical-documentation-page.netlify.app/</a>	0.07%
58	<b>studopedia.net</b> <a href="https://studopedia.net/3_24015_karta-inzhenerno-geologichnogo-rayonuvannya.html">https://studopedia.net/3_24015_karta-inzhenerno-geologichnogo-rayonuvannya.html</a>	0.07%
59	<b>www.BiblioFond.ru</b> <a href="https://www.BiblioFond.ru/view.aspx?id=731708">https://www.BiblioFond.ru/view.aspx?id=731708</a>	0.07%
60	<b>revolution.allbest.ru</b> <a href="https://revolution.allbest.ru/audit/00560801_0.html">https://revolution.allbest.ru/audit/00560801_0.html</a>	0.07%
61	<b>www.geokniga.org</b> <a href="https://www.geokniga.org/bookfiles/geokniga-transportni-sistemi-girnichih.pdf">https://www.geokniga.org/bookfiles/geokniga-transportni-sistemi-girnichih.pdf</a>	0.07%
62	<b>otherreferats.allbest.ru</b> <a href="https://otherreferats.allbest.ru/management/00006546_0.html">https://otherreferats.allbest.ru/management/00006546_0.html</a>	0.07%
63	<b>www.miireferat.com</b> <a href="https://www.miireferat.com/poshuk/%D1%80%D0%B8%D0%BD%D0%BA%D0%BE%D0%B2%D0%B8%D0%B9-%D0%BF%...">https://www.miireferat.com/poshuk/%D1%80%D0%B8%D0%BD%D0%BA%D0%BE%D0%B2%D0%B8%D0%B9-%D0%BF%...</a>	0.07%
64	<b>otherreferats.allbest.ru</b> <a href="https://otherreferats.allbest.ru/audit/00080840_0.html">https://otherreferats.allbest.ru/audit/00080840_0.html</a>	0.07%
65	<b>myleksii.su</b> <a href="https://myleksii.su/7-43511.html">https://myleksii.su/7-43511.html</a>	0.07%
66	<b>nashaucheba.ru</b> <a href="https://nashaucheba.ru/v17106/%D0%B4%D0%BE%D0%B2%D1%96%D0%B4%D0%BD%D0%B8%D0%BA_%D0%BC.%D...">https://nashaucheba.ru/v17106/%D0%B4%D0%BE%D0%B2%D1%96%D0%B4%D0%BD%D0%B8%D0%BA_%D0%BC.%D...</a>	0.07%
67	<b>upwikihir.top</b> <a href="https://upwikihir.top/wiki/HTML">https://upwikihir.top/wiki/HTML</a>	0.06%

# Диплом.СельвесюкІ.І.

Завантажено: 06/19/2022 | Перевірено: 06/19/2022

● Схожість ● Цитата ● Використані джерела ● Заміна символів

68	<b>upwikiro.top</b> <a href="https://upwikiro.top/wiki/HTML">https://upwikiro.top/wiki/HTML</a>	0.06%
69	<b>butanasi.github.io</b> <a href="https://butanasi.github.io/work-study/index.html">https://butanasi.github.io/work-study/index.html</a>	0.06%
70	<b>www.ijraset.com</b> <a href="https://www.ijraset.com/files/serve.php?FID=21720">https://www.ijraset.com/files/serve.php?FID=21720</a>	0.06%
71	<b>wikizero.com</b> <a href="https://wikizero.com/en/IntelliJ_IDEA">https://wikizero.com/en/IntelliJ_IDEA</a>	0.06%
72	<b>en.wikipedia.org</b> <a href="https://en.wikipedia.org/wiki/IntelliJ_IDEA">https://en.wikipedia.org/wiki/IntelliJ_IDEA</a>	0.06%
73	<b>virtual-dba.com</b> <a href="https://virtual-dba.com/blog/database-101-popular-databases-mysql/">https://virtual-dba.com/blog/database-101-popular-databases-mysql/</a>	0.06%
74	<b>nwikiw.cyou</b> <a href="https://nwikiw.cyou/wiki/HTML">https://nwikiw.cyou/wiki/HTML</a>	0.06%
75	<b>mdbootstrap.com</b> <a href="https://mdbootstrap.com/docs/standard/extended/login/">https://mdbootstrap.com/docs/standard/extended/login/</a>	0.06%
76	<b>siam.press</b> <a href="https://siam.press/wiki/ru/IntelliJ">https://siam.press/wiki/ru/IntelliJ</a>	0.06%
77	<b>wikibgu.ru</b> <a href="https://wikibgu.ru/wiki/HTML">https://wikibgu.ru/wiki/HTML</a>	0.05%
78	<b>vvpusp.vn.ua</b> <a href="https://vvpusp.vn.ua/sites/default/files/fields/paragraph.document.field_document_file/2020-02/Informatyka-11-klas-Ruden...">https://vvpusp.vn.ua/sites/default/files/fields/paragraph.document.field_document_file/2020-02/Informatyka-11-klas-Ruden...</a>	0.05%
79	<b>amac.md</b> <a href="http://amac.md/Biblioteca/data/28/14/05/02/36.2.pdf">http://amac.md/Biblioteca/data/28/14/05/02/36.2.pdf</a>	0.05%
80	<b>eprints.library.odeku.edu.ua</b> <a href="http://eprints.library.odeku.edu.ua/5937/1/Nebozh_Vitaly_Validaciya_kviv_B2019.pdf">http://eprints.library.odeku.edu.ua/5937/1/Nebozh_Vitaly_Validaciya_kviv_B2019.pdf</a>	0.05%
81	<b>studall.org</b> <a href="https://studall.org/all2-23232.html">https://studall.org/all2-23232.html</a>	0.05%
82	<b>uk.wikipedia.org</b> <a href="https://uk.wikipedia.org/wiki/XML">https://uk.wikipedia.org/wiki/XML</a>	0.05%
83	<b>www.marganets-proflyceum.dp.sch.in.ua</b> <a href="http://www.marganets-proflyceum.dp.sch.in.ua/Files/downloadcenter/Informatyka-11-klas-Rudenko-2019.pdf">http://www.marganets-proflyceum.dp.sch.in.ua/Files/downloadcenter/Informatyka-11-klas-Rudenko-2019.pdf</a>	0.05%
84	<b>antibotan.com</b> <a href="https://antibotan.com/file.html?work_id=111985">https://antibotan.com/file.html?work_id=111985</a>	0.05%
85	<b>lib.iitta.gov.ua</b> <a href="https://lib.iitta.gov.ua/719887/1/26_5.pdf">https://lib.iitta.gov.ua/719887/1/26_5.pdf</a>	0.05%
86	<b>revolution.allbest.ru</b> <a href="https://revolution.allbest.ru/law/01261132_0.html">https://revolution.allbest.ru/law/01261132_0.html</a>	0.05%

# Диплом.СельвесюкІ.І.

Завантажено: 06/19/2022 | Перевірено: 06/19/2022

● Схожість ● Цитата ● Використані джерела ● Заміна символів

87	<b>revolution.allbest.ru</b> <a href="https://revolution.allbest.ru/law/00598778_0.html">https://revolution.allbest.ru/law/00598778_0.html</a>	0.05%
88	<b>revolution.allbest.ru</b> <a href="https://revolution.allbest.ru/bank/00245249_0.html">https://revolution.allbest.ru/bank/00245249_0.html</a>	0.05%
89	<b>docplayer.net</b> <a href="https://docplayer.net/85895600-Meteorologichni-priladi-i-metodi-sposterezhen.html">https://docplayer.net/85895600-Meteorologichni-priladi-i-metodi-sposterezhen.html</a>	0.05%
90	<b>nashaucheba.ru</b> <a href="https://nashaucheba.ru/v45206/?cc=1">https://nashaucheba.ru/v45206/?cc=1</a>	0.05%
91	<b>www.rostlab.org</b> <a href="https://www.rostlab.org/owiki/images/4/44/Mysql_slides.pdf">https://www.rostlab.org/owiki/images/4/44/Mysql_slides.pdf</a>	0.05%

## Цитати

### Включені цитати

10

- 1 «Розробка Веб-сервісу для інформаційної підтримки діяльності агентства зна й омств »
- 2 "Інформаційні послуги"
- 3 «фахівців з інформаційних систем, орієнтованих на інтеграцію інформаційно-технологічних рішень та бізнес-процесів для задоволення інформаційних потреб бізнесу та інших підприємств»
- 4 «не тільки користувачі, а й ті, хто керує та обслуговує комп'ютери, ті, хто підтримує дані, і ті, хто підтримує комп'ютерні мережі»
- 5 "веб-додатками"
- 6 «сервлет HTTP»
- 7 «інтерфейси розробника є, а документація (не кажучи вже про зворотний зв'язок у реальному світі через веб-сайти тощо) дуже, дуже добре"
- 8 «швидкий, стабільний і справжній багатокористувацький, багатопоточний сервер баз даних SQL»
- 9 "схеми використання - це принципи вашої системи"
- 10 " робить система (а не "

## Використані джерела

1. Stepen Jones. "New Matrimonial Plan", The Spirit of the Public Journals for 1799: Being an Impartial Selection of the Most Exquisite Essays and Jeux d'Esprits, Particularly Prose, That Appear in the Newspapers and Other Publications: **Видання**. Palala Press, 2008. 329 с 2. William Hone. "A New Matrimonial Plan", The Every-day Book and Table Book: **Видання**. London, 1838, 22-23 сс. 3. FBI Cautions Public to be Wary of Online Romance Scams. URL: <https://www.houstonpublicmedia.org/articles/news/criminal-justice/2022/02/11/418897> (дата звернення 01.06.2022). 4. Marriage agency - Wiktionary. URL: [en.wiktionary.org](https://en.wiktionary.org) (дата звернення 01.06.2022). 5. The evolution of: matchmaking - Reader's Digest. URL: [www.readersdigest.co.uk](https://www.readersdigest.co.uk) (дата звернення 01.06.2022). 6. Social and Demographic Trends Team . URL: <https://www.pewresearch.org/about/research-teams/social-and-demographic-trends-team/> (дата звернення 01.06.2022). 7. New Partners, More Kids: Multiple-Partner Fertility in the United States URL: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4182921/12-14> (дата звернення 01.06.2022). 8. America Is a Singles Nation, Or Is It? URL: <https://www.psychologytoday.com/us/blog/living-single/201506/america-is-singles-nation-or-is-it> (дата звернення 01.06.2022). 9. NowUKnow: Why Millennials Refuse to Get Married | Bentley University. URL: [www.bentley.edu](https://www.bentley.edu) (дата звернення 01.06.2022). 10. Japan's Unmarried Masses Face Mounting Obstacles to Matrimony. URL: [www.nippon.com](https://www.nippon.com) (дата звернення 01.06.2022). 11. Jackson, Molly. "How Americans' views of marriage have changed over time". URL: <https://www.businessinsider.com/how-americans-views-of-marriage-have-changed-over-time-2016-2> (дата звернення 01.06.2022). 12. Lichter, Daniel T.; Price, Joseph P.; Swigert, Jeffrey M. Mismatches in the Marriage Market. Journal of Marriage and Family: **Видання**. Lichter, Daniel T., 2020, 509-514. 13. Are Dating Apps Damaging Our Mental Health? URL: <https://www.psychologytoday.com> (дата звернення 01.06.2022). 14. Dating Agency. URL: <https://bearl.one/> (дата звернення 01.06.2022). 15. Message from CEO | IBJ Inc. URL: [www.ijb-gl.com](https://www.ijb-gl.com) (дата звернення 01.06.2022). 16. National Institute of Population, Marriage Process and Fertility of Married Couples Attitudes toward Marriage and Family among Japanese Singles: **Статистичні дані**. National Institute of Population, 2017, 27 с. 17. Ryan W. Buell, Michael I. Norton. The Labor Illusion: How Operational Transparency Increases Perceived Value: **Наукова публікація**. Harvard Business School, Harvard University, Boston, Massachusetts, 2017, 1564-1579 сс 18. The "Ehime Method": Using Big Data to Support Matchmaking. URL: [nippon.com](https://www.nippon.com) (дата звернення 01.06.2022). 19. Marriage Abroad. URL: [travel.state.gov](https://travel.state.gov). (дата звернення 01.06.2022). 20. Getting married abroad. URL: [GOV.UK](https://gov.uk) (дата звернення 01.06.2022). 21. Marriage Agency | Marriage Matching Marriage Agency. URL: <https://marriagematching.love/marriage-agency/> (дата звернення 01.06.2022). 22. "International Marriage Agency - ELITE. URL: [www.magzter.com](https://www.magzter.com). (дата звернення 01.06.2022). 23. 8 facts about love and marriage in America. URL: <https://www.pewresearch.org/fact-tank/2019/02/13/8-facts-about-love-and-marriage/> (дата звернення 01.06.2022). 24. Murphy, Hannah. Virtual dating on the rise as users seek lockdown love, says Match chief. URL: <https://www.ft.com/content/f970c353-65e3-4515-a179-e94b89ce3775> (дата звернення 01.06.2022). 25. Marriage agency URL: [https://en.everybodywiki.com/Marriage\\_agency](https://en.everybodywiki.com/Marriage_agency) (дата звернення 01.06.2022). 26. **Брайан Глік. Писати один раз, бігти куди завгодно? Комп'ютерний тижневик**. URL: <https://www.computerweekly.com/feature/Write-once-run-anywhere> (дата звернення 01.06.2022). 27. The Java Language Environment URL: <https://www.oracle.com/java/technologies/introduction-to-java.html> (дата звернення 01.06.2022) 28. **Чан Розалі. 10 найпопулярніших мов програмування за версією Facebook для програмістів** URL: <https://www.businessinsider.de/the-10-most-popular-programming-languages-according-to-github-2018-10-op=1>. (дата звернення 01.06.2022) 29. **Писати один раз, бігти куди завгодно? Комп'ютерний тижневик**. URL: <https://www.computerweekly.com/feature/Write-once-run-anywhere>. (дата звернення 01.06.2022) 30. **Конспект лекцій з навчальної дисципліни «Розподілені сервісні системи» для студентів всіх форм навчання спеціальності «Телекомунікації та радіотехніка» відділу Інфокомунікацій та інженерії. / Уклад.: О.Д. Архелюк - Чернівці.: ЧНУ імені Юрія Федьковича 2021. - 94 с.** 31. **Єлович, Деян. Чому Java завжди буде повільніше, ніж C ++.** URL: [https://uk.wikicsu.ru/wiki/java\\_\(programming\\_language\)](https://uk.wikicsu.ru/wiki/java_(programming_language)). (дата звернення 01.06.2022) 32. Java 2 Platform Enterprise Edition. URL: <https://www.oracle.com/java/technologies/>. (дата звернення 01.06.2022) 33. **Вибір першої мови програмування — огляд Java.** URL: <https://blog.ithillel.ua/articles/vybir-pershoi-movy-prohramuвання-ohljad-java>. (дата звернення 01.06.2022). 34. **Копитко М.Ф., Іванків К.С. Основи програмування мовою Java: Тексти лекцій. - Львів: Видавничий центр ЛНУ ім. Івана Франка, 2002.- 83 с.** 35. IntelliJ IDEA :: рефакторинг Java плюс складний рефакторинг коду для JSP, XML, CSS, HTML, JavaScript. URL: <https://www.jetbrains.com/idea/features/refactoring.html>. (дата звернення 01.06.2022). 36. **К.Арнольд, Дж.Гослінг. Мова програмування Java/Пер.с англ.-Санкт- Петербург, 1997.-304 с.** 37. **Керування пам'яттю у Java Script.** URL: <https://codeguida.com/post/1045>. (дата звернення 01.06.2022). 38. **К.Арнольд, Дж.Гослінг. Мова програмування Java/Пер.с англ.-СПб: Пітер, 1997.-304 с.** 39. What is Memcached? URL: <http://memcached.org/> дата звернення 01.06.2022). 40. **Методичні вказівки до лабораторних робіт "Створення веб-сайту за допомогою HTML" з курсу "Основи Internet-технологій" : для студ. спец. 124 "Системний аналіз", 186 "Видавництво і поліграфія" / уклад. А. Н. Марченко, В. О. Колбасін, В. П. Прокопенков ; Харківський політехнічний ін-т, нац. техн. ун-т. - Харків : НТУ "ХПІ", 2017. - 67 с.**

# Диплом.СельвесюкІ.І.

Завантажено: 06/19/2022 | Перевірено: 06/19/2022

● Схожість ● Цитата ● Використані джерела ● Заміна символів

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**Національний університет водного господарства та природокористування**  
**Навчально-науковий інститут автоматичної, кібернетичної та**  
**обчислювальної техніки**  
**Кафедра комп'ютерних технологій та економічної кібернетичної**

**Допущено до захисту:**

Завідувач кафедри

\_\_\_\_\_ д. е. н., проф. П. М. Грицюк

« \_\_\_\_\_ » \_\_\_\_\_ 2022 р.

## **КВАЛІФІКАЦІЙНА РОБОТА**

на здобуття ступеня «бакалавр»

за освітньо-професійною програмою «Інформаційні системи і технології»  
спеціальності 126 «Інформаційні системи та технології»

на тему: **«Розробка Веб-сервісу для інформаційної підтримки діяльності**  
**агентства з на**

**Виконав:**

здобувач вищої освіти 4 курсу, групи ІСТ-41  
**Сельвесюк Ірина Ігорівна**

**Керівник:**

канд. техн. наук, доцент Барановський С. В.

**Рецензент:**

канд. техн. наук, доцент Карпович І. М.

Рівне – 2022

1

## ЗМІСТ

ВСТУП.....	
РОЗДІЛ 1 АНАЛІЗ СУЧАСНОГО СТАНУ ПИТАННЯ.....	5
1.1 Характеристика та особливості управління діяльністю агентства знайомств.....	
1.2 Інформаційна система підприємства з надання послуг.....	11
1.3 Сучасні тенденції веб-розробки.....	15
1.4 Аналіз існуючих рішень.....	28
1.5 Постановка задачі.....	29
РОЗДІЛ 2 АНАЛІЗ ІНСТРУМЕНТАЛЬНИХ ЗАСОБІВ РЕАЛІЗАЦІЇ.....	31
2.1 Огляд мови програмування.....	31
2.1 Огляд середовища розробки.....	38
2.3 Огляд додаткового інструментарію.....	39
РОЗДІЛ 3 ПРОЕКТУВАННЯ І РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ...53	
3.1 Аналіз варіантів використання.....	53
3.2 Проектування внутрішньої будови.....	55
3.2 Розробка графічного інтерфейсу системи.....	61
ВИСНОВКИ.....	
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	70

2

## ВСТУП

В сучасному світі кожного дня кількість інформації, яку необхідно зберігати росте в геометричній прогресії, набуваючи величезних розмірів. В певний момент старі варіанти зберігання інформації, такі як паперові носії, або навіть усна форма, віджили своє і більше не могли задовольняти потреби сучасного соціуму.

Саме в цей момент і з'явилося систем автоматизації, які до сьогодення допомагають людям у структуруванні та зберіганні інформації різного роду, взаємодії між ролями користувачів, тощо.

На сьогоднішній день кожне підприємство та установа мають свої внутрішні клієнти, які допомагають зберігати і структурувати інформацію про роботи даної установи, автоматизувати деякі процеси, тощо.

Виходячи з актуальності явища систем автоматизації, об'єктом дослідження є інформаційні системи, що забезпечують автоматизацію бізнес-процесів підприємства та інформаційний сервіс, і їх використання в різного роду структурах. Інформаційні системи – це комбінації апаратних, програмних і телекомунікаційних мереж, які люди створюють і використовують для збору, створення та поширення корисних даних. Інформаційні системи оточують вас майже щодня. Чудовим прикладом є мережі Wi-Fi в університеті, служби пошуку бази даних у навчальному центрі та принтери в комп'ютерних лабораторіях. Кожен раз, коли ви ходите по магазинах, ви взаємодієте з інформаційною системою, яка керує запасами та продажами. Кожен смартфон постійно підключається до доступних мереж через Wi-Fi, записуючи ваше місцезнаходження та інші дані.

Предмет дослідження — система контролю процесів агентства знайомств. Управління процесом – це здатність контролювати та коригувати процес для отримання бажаного результату. Він використовується в промисловості та бізнесі для підтримки якості та підвищення продуктивності. Отримавши доступ до ізольованої інформації, бізнес-

3

послуги можуть розвиватися; придбати не тільки детальний уявлення про взаємодію їхніх процесів, але також отримувати зворотній зв'язок у реальному часі з служб фізичного домену і гнучко взаємодіяти з ними.

Мета роботи — розроблення власної реалізації системи контролю процесів агентства знайомств.

Для досягнення поставленої мети слід виконати наступні завдання:

- Провести огляд діяльності агентства знайомств;
- Провести огляд інформаційної системи підприємства з надання послуг;
- Провести аналіз методів та засобів розробки веб-сайтів;
- Провести огляд існуючих рішень;
- Обрати мову програмування;
- Обрати додаткові інструменти;
- Обрати середовище розробки;
- Провести аналіз варіантів використання;
- Провести проектування внутрішньої будови;
- Розробити графічний інтерфейс користувача;

Методологічну основу роботи складає сукупність структурних методів, використаних для вирішення поставлених завдань, а саме: послідовна деталізація системи, використання графічного подання взаємозв'язків елементів системи; для реалізації практичної частини було застосовано методологію створення інформаційних систем, а саме: аналіз ІС на основі процесів, які протікають в організації.

Структура роботи включає в себе:

- аналіз тенденцій розвитку веб-розробки інформаційних систем та дослідження сучасних підходів до розробки та впровадження інформаційних систем на підприємствах, що дозволяє оцінити актуальність роботи;
- аналіз інструментальних засобів розробки, що дозволяє ознайомитись детальніше з мовою програмування так середовищем

4

розробки використаним для створення для інформаційної підтримки діяльності агентства знайомств;

- детальний опис проектування і розробки програмного забезпечення, який демонструє розроблену систему, функції, проектування та варіації використання для потенційного користувача.

## РОЗДІЛ 1

### АНАЛІЗ СУЧАСНОГО СТАНУ ПИТАННЯ

#### 1.1 Характеристика та особливості управління діяльністю агентства знайомств

Агентство знайомств, також відоме як шлюбне бюро, шлюбне агентство, подружнє бюро або подружнє агентство, — це компанія, яка надає послуги зі знайомства потенційним парам з метою встановлення комунікації між ними відповідно до мети спілкування кожного клієнта.

Варіації діяльності агентств знайомств:

- Зустріч віч-на-віч: чоловіки та жінки приходять особисто і просять свати допомогти їм знайти потенційного партнера.
- Інтернет-агентство знайомств: веб-сайт, на якому користувачі реєструються, публікують свої профілі та зв'язуються з іншими учасниками, які зареєструвалися в агентстві.
- Швидкі знайомства: група людей чергує партнерів і описує їх особистість і бажання протягом встановленого часу.
- Шлюбне агентство: агентство, яке спеціально допомагає людям знайти шлюбного партнера, а не когось для випадкових чи серйозних побачень. Вони можуть носити внутрішній або міжнародний характер.

Інтернет та агенції швидких знайомств є найбільшими в групі.

Зростає кількість компаній, які навчають чоловіків, як вони можуть самі зустрічатися з жінками без використання агентства знайомств, деякі з яких використовують лейбл pickup artist.

І агенції знайомств, і шлюбні агентства мають деякі організації, де людям платять за взаємодію з іншими членами та підтримання їхнього інтересу. Хоча існують різні агенції, які мають лише членів, які справді шукають справжніх партнерів.

6

Щоб зрозуміти актуальність використання послуж шлюбних агентств на сьогоднішній день, варто також заглибитись в історію їх виникнення та використання в минулі часи.

Шлюбні агентства, якими керували священнослужителі, були запроваджені в Англії та Уельсі наприкінці 18 століття, що викликало значну розвагу у соціальних коментаторів того часу. У 1799 році в «провінційному виданні» йдеться про те, що Шлюбний план пропонується створити в кожному графстві, місті чи містечку, в Англії чи Уельсі. Система цього цікавого, і це має здатися насправді серйозним, плану — наскільки ми можемо дізнатися, — це таке: — Кожна особа будь-якої статі, яка бажає укласти шлюбний договір, спочатку підписує певну суму. Усі пані та джентльмени повинні описати себе справжніми чи вигаданими іменами, як вони хочуть».[1]

Хоча більшість людей зустрічаються в громадських організаціях, у повсякденному житті та на роботі або знайомляться через друзів чи родичів, комерційні агенції знайомств сильно, але непомітно з'явилися в західному світі після Другої світової війни, в основному обслуговуючи людей 25–44 років. вікова група. Поширеними стали також персональні оголошення в газетах і журналах.

З моменту появи Інтернету пошук партнерів і залицяння зазнали змін завдяки сервісам онлайн-знайомств і мобільних послуг знайомств. Приблизно з 1995 року телекомунікації та комп'ютерні технології швидко розвивалися, дозволяючи знайомим використовувати домашні телефони з автовідповідачами – мобільними телефонами – та веб-системи для пошуку потенційних партнерів. «Попередні побачення» можуть відбутися по телефону або онлайн за допомогою миттєвих повідомлень, електронної пошти чи навіть відеозв'язку.

Багато одиноких людей шукають кохання в Інтернеті, а дослідження у Сполученому Королівстві показують, що станом на 2004 рік у цій країні було

7

близько 150 агентств, де ринок, очевидно, зростає приблизно на 20 відсотків на рік. Академічні дослідники вважають неможливим знайти точні цифри щодо важливих статистичних даних, таких як співвідношення активних знайомих і великої кількості неактивних членів (яких агентство часто помилково називає потенційними партнерами, залишаючи їх «на обліку» ще довго після того, як вони ліворуч) і загальне співвідношення чоловіків і жінок у агенції. Академічні дослідження традиційних агентств до Інтернету показують, що більшість таких агенцій мали набагато більше чоловіків, ніж жінок у своєму членстві. Через співвідношення наявних одиноких жінок, упереджених проти чоловіків у західному світі, багато агентств знайомих і шлюбів почали працювати над багатьма агенціями знайомих і шлюбів. пропонувати послуги за кордоном. Традиційно в багатьох суспільствах (включаючи західні) роль переслідувача виконуватимуть чоловіки. Однак анонімність Інтернету (а також інші фактори) дозволила жінкам взяти на себе цю роль онлайн.

Тенденція одиноких людей, які підключаються до Інтернету, продовжує зростати, оскільки відсоток неодружених Північної Америки, які спробували зустрітися в Інтернеті, зріс з двох відсотків у 1999 році до більш ніж десяти відсотків сьогодні (з Canadian Business, лютий 2002 року). Більше половини онлайн-споживачів (53%) знають когось, хто зав'язав дружбу чи стосунки в Інтернеті, а три чверті онлайн-покупців у віці від 18 до 24 років (74%) стверджують, що знають. Існують також деякі академічні докази того, що вікова група 18–25 років значною мірою зайнялася онлайн-знайомствами.

Шлюбні агентства еволюціонували з агентств знайомих. Історично їхні процеси походять від процесів сватання, які сягають 1600-х років в Англії, коли парафіяльні вікарії відігравали важливу роль у поєднанні своїх парафіян із подружжям із того самого соціального класу.[5] Подібна практика використовувалася в Японії з 16 століття, коли Надоко (свати) виконувала обов'язки Омій (введення в шлюб) для батьків, які шукали гідного чоловіка

8

для своєї дитини. В Англії релігійні зв'язки були розірвані в 1825 році з появою агенцій знайомств у Лондоні, що стало новим варіантом знайти когось підходящого для тих, хто зазнав труднощів у зустрічі з шлюбним партнером.

Зміни в культурі та суспільстві призвели до зміни рівня шлюбності через погляди одиноких людей на шлюб. У 1960 році близько 12% дорослих у віці від 25 до 34 років ніколи не були одружені. Через 10 років, коли тій самій групі було від 35 до 44 років, 7% з них все ще були неодружені. Ще 10 років потому, у 1980 році, коли цим людям було від 45 до 54 років, 5% ще не були одружені. Наступна група, починаючи з 1970 року, пішла подібним шляхом. [6] У 1960 році коефіцієнт шлюбності становив 72% для всіх дорослих віком від 18 років, у 2010 році коефіцієнт шлюбів становив 51% для тієї ж демографічної групи.[7]

Повідомляється, що частина розриву в шлюбі[8] пов'язана з тим, що люди чекають, поки вони стануть старшими, отримають вищу освіту та стануть більш фінансово стабільними.[9] Оскільки ті, хто чекав, одружуються з іншими професійно успішними людьми, їхнє бажання вказується як бажання передати своїм дітям як освітній, так і фінансовий капітал, і забезпечити стабільність для своїх дітей.

На додаток до цього в різних розвинених країнах спостерігається зростання кількості тих, хто просто хоче залишитися самотнім і мати свободу[10] або пережити те, що називається відстроченим підлітковим віком.[11] Відчутне зниження лояльності партнера, низький рівень самооцінки, сильне бажання фінансової стабільності, проблеми у стосунках і зростання тенденції розлучень серед подружніх пар, часто викликають у багатьох сумнівів, чи будуть партнери, яких вони зустрічають, підходящими партнерами для шлюбу. [12] Крім того, користувачі додатків для знайомств посилаються на проблеми довіри, які виникають через обман і нечесні наміри у стосунках, які блокують їх розумово та емоційно від розвитку серйозних

9

стосунків, які можуть призвести до шлюбу з людьми, з якими познайомилися через програми для знайомств.[13]

Сучасне шлюбне агентство працює двома принципами, щоб допомогти клієнтам знайти шлюбного партнера:

- Маючи групу членів, які спеціально шукають шлюбних партнерів.
- Сватання.

Історично шлюбні агенції зазвичай працювали окремо, сучасні шлюбні агенції зараз часто працюють в рамках федерацій або асоціацій[14], що забезпечує ефект масштабу, який дозволяє легше створювати шлюбні збіги завдяки співпраці між шлюбними агентствами. Це поширене явище в Японії, де культура разом із соціальним впливом[15] призвели до того, що шлюбні агентства були відповідальними за 5,3% шлюбів у 2010 та 2014 роках.[16]

Сватання є основною діяльністю для шлюбних агентств, оскільки їхні члени зосереджені на тому, щоб бути з кимось, кого вони справді люблять і з ким справді сумісні, завдяки підвищенню рівня щастя та самореалізації.[17]

Шлюбні агентства використовують сватання, щоб забезпечити найкращі шанси на успіх шлюбів у довгостроковій і короткостроковій перспективі, проблеми пов'язані з тим, наскільки кваліфіковані свати, а також від того, наскільки добре вони знають своїх членів. У той час як сватання раніше використовувалися в усьому світі для забезпечення міцних союзів між сім'ями, сучасні шлюбні агентства приділяють більше уваги психологічному аспекту того, як формується любов між двома людьми та їх придатності один для одного як супутників життя. Через це шлюбні агентства проводять сватання, щоб оцінити сумісність перед знайомством. Цей тип практики був поширеним в Японії з 1940-х років, коли система вступу в шлюб Оміаї була змінена з сімейного походження на створення любовних пар для шлюбу. Підбір сватів або шлюбів, коли це здійснюється шлюбними агентствами, виконується вручну, а не за комп'ютерним алгоритмом, який є звичайним для програм для знайомств.

10

У той час як підбір сватів у додатках для знайомств працює зі зваженими змінними, отриманими з профілів, шлюбні агентства вручну підбирають контакти в результаті того, що співробітники шлюбного агентства розвивають розуміння особистості та бажань своїх клієнтів, часто в поєднанні з використанням великих даних[18] з пошуків колективні бази даних, які зберігаються між партнерськими шлюбними агентствами в межах їхньої федерації чи асоціації.

Реєстрація в шлюбному агентстві вимагає підтвердження права на шлюб, при цьому шлюбні агентства зазвичай вимагають свідоцтво про відсутність перешкод для шлюбу, свідоцтво про відсутність записів про шлюб, свідчення про неодружений статус або інший подібний юридичний доказ здатності заявника одружитися. Хоча така документація є загальною для людей, які одружуються за кордоном, [19][20] шлюби в одній і тій же країні є нетиповими у багатьох західних країнах.

Послуги шлюбного агентства, такі як коучинг, щоб допомогти клієнтам подолати такі проблеми, як нервозність, занепокоєння та психологічні проблеми, викликані попередніми стосунками, стають все більш поширеними, щоб допомогти їм досягти стабільних любовних стосунків. Шлюбні агенції, на відміну від агенцій знайомств, також повністю зосереджені на тому, щоб їхні клієнти одружилися, а не просто були супутниками життя,[21] знайомствами, співжиттям або створенням інших корисних відносин.

Широкої популярності також набули міжнародні шлюбні агенції, які відрізняються від послуг з замовлення нареченої поштою, дозволяють клієнтам шукати свого ідеального шлюбного партнера по всьому світу.

Використання міжнародних шлюбних агентств підживлювалося збільшенням знайомства з іноземними культурами в усьому світі. Оскільки багато в західних країнах зараз шукають шлюбних партнерів із більш традиційних східних культур, особливо в Японії, оскільки культурні цінності

11

вірності, доброти та любові є основними цінностями, яких насправді шукають ті, хто насправді бажає одружитися, але їм важко знайти у своїх країнах. [22] Такі специфічні типи шлюбних агентств, наприклад, японські шлюбні агенції, виникли для того, щоб задовольнити бажання тих, хто шукає шлюбу з кимось із певного географічного регіону чи культурної системи.

Збільшення числа міжнародних шлюбних агентств також відображає тенденцію зростання змішаних шлюбів, коли партнери з різним етнічним або расовим походженням вступають у шлюб. У США лише 3% нових шлюбів були класифіковані як змішані шлюби в 1967 році, до 2015 року змішані шлюби зросли до 17%, при цьому азіати становили 29% і латиноамериканці 27% цієї цифри[23].

## 1.2 Інформаційна система підприємства з надання послуг

Ефективність діяльності агентств знайомств в сучасних умовах значною мірою визначається рівнем застосування сучасних інформаційних технологій та інформаційних систем. Таким чином, поряд з автоматизацією робочого процесу, існує великий попит на інструменти управління бізнес-процесами за допомогою інформаційної системи.

Інформаційна система (ІС) - це формальна, соціально-технічна, організаційна система, призначена для збору, обробки, зберігання та розповсюдження інформації. [1] У соціотехнічній перспективі інформаційні системи складаються з чотирьох компонентів: завдання, людей, структури (або ролей) та технології [32]. Інформаційні системи можна визначити як інтеграцію компонентів для збору, зберігання та обробки даних, дані яких використовуються для надання інформації, сприяння знанням, а також цифрових продуктів. [3]

Комп'ютерна інформаційна система - це система, що складається з людей та комп'ютерів, яка обробляє або інтерпретує інформацію. [34] Цей

12

термін також іноді використовують для простого позначення комп'ютерної системи з встановленим програмним забезпеченням.

Інформаційні системи - це академічне дослідження систем із конкретним посиланням на інформацію та додаткові мережі апаратного та програмного забезпечення, які люди та організації використовують для збору, фільтрування, обробки, створення та розповсюдження даних. Акцент робиться на інформаційній системі, що має певні межі, користувача, процесорах, сховищах, входах, виходах та згаданих мережах зв'язку [8].

У багатьох організаціях відділ або підрозділ, відповідальний за інформаційні системи та обробку даних, відомий як "Інформаційні послуги". [9]

Будь-яка конкретна інформаційна система спрямована на підтримку операцій, управління та прийняття рішень. [13] Інформаційна система - це інформаційно-комунікаційні технології (ІКТ), які використовує організація, а також спосіб взаємодії людей із цією технологією для підтримки бізнес-процесів [15].

Деякі автори чітко розмежовують інформаційні системи, комп'ютерні системи та бізнес-процеси. Інформаційні системи, як правило, включають компонент ІКТ, але вони не стосуються лише ІКТ, а натомість зосереджуються на кінцевому використанні інформаційних технологій. Інформаційні системи також відрізняються від бізнес-процесів. Інформаційні системи допомагають контролювати ефективність бізнес-процесів. [16]

Альтер [17] аргументує переваги розгляду інформаційної системи як особливого типу робочої системи. Робоча система - це система, в якій люди або машини виконують процеси та дії, використовуючи ресурси для виробництва конкретних продуктів чи послуг для клієнтів. Інформаційна система - це робоча система, діяльність якої спрямована на збір, передачу, зберігання, пошук, маніпулювання та відображення інформації. [19]

13

Як такі, інформаційні системи взаємопов'язані з системами даних, з одного боку, та системами діяльності, з іншого. [20] Інформаційна система - це форма комунікаційної системи, в якій дані представляють і обробляються як форма соціальної пам'яті. Інформаційну систему також можна вважати напівформальною мовою, яка підтримує прийняття рішень і дій людини. Інформаційні системи є основною сферою досліджень організаційної інформатики. [21]

Сільвер (1995) пропонує дві точки зору на інтелектуальну власність, включаючи програмне забезпечення, апаратне забезпечення, дані, людей і програми. [22] Чжен надає інший системний погляд на інформаційну систему, який також додає процеси та ключові елементи системи, такі як контекст, кордони, цілі та взаємодії.

Асоціація обчислювальної техніки визначає «фахівців з інформаційних систем, орієнтованих на інтеграцію інформаційно-технологічних рішень та бізнес-процесів для задоволення інформаційних потреб бізнесу та інших підприємств» [23].

Існують різні типи інформаційних систем, наприклад системи обробки транзакцій, системи підтримки прийняття рішень, системи управління знаннями, системи управління навчанням, системи управління базами даних та офісні інформаційні системи. Ключем до більшості інформаційних систем є інформаційні технології, які часто призначені для того, щоб люди могли виконувати завдання, для яких людський мозок не підходить, наприклад, обробляти великі обсяги інформації, виконувати складні обчислення та керувати багатьма одночасними процесами.

Інформаційні технології є дуже важливим і гнучким ресурсом, доступним для менеджерів. [24] Багато компаній заснували посаду головного інформаційного директора (CIO), який працює у виконавчому комітеті разом із головним виконавчим директором (CEO), фінансовим директором (CFO), головним операційним директором (COO) та технологом. (головний

14

технічний директор). СТО також може виконувати роль СІО і навпаки. Генеральний директор з інформаційної безпеки (CISO) зосереджується на управлінні інформаційною безпекою.

Для створення інформаційної системи необхідно об'єднати шість компонентів:

- **Обладнання:** Термін обладнання відноситься до машин та обладнання. У сучасних інформаційних системах до цієї категорії відноситься сам комп'ютер і всі його аксесуари. До допоміжного устаткування відноситься обладнання введення і виведення, обладнання зберігання та обладнання зв'язку. В інформаційних системах до комп'ютерів апаратне забезпечення могло складатися з книг і чорнила.
- **Програмне забезпечення:** термін програмне забезпечення відноситься до комп'ютерних програм і посібників (якщо є), які їх підтримують. Комп'ютерні програми – це машиночитані інструкції, які спрямовують схеми в апаратному забезпеченні системи на роботу таким чином, щоб отримати корисну інформацію з даних. Програми зазвичай зберігаються на певному носії введення-виводу, як правило, на диску або стрічці. «Програмне забезпечення» передкомп'ютерної інформаційної системи включає в себе те, як підготувати пристрій до використання (наприклад, заголовки стовпців у книзі) та інструкції з використання (керівники з карткового каталогу).
- **Дані:** дані – це факт, який система використовує для отримання корисної інформації. У сучасних інформаційних системах дані зазвичай зберігаються на диску або стрічці в машиночитаній формі, доки комп'ютеру це не знадобиться. В інформаційних системах до появи комп'ютерів дані зазвичай зберігалися в доступній для читання формі.
- **Процедури:** Процедури – це політики, які регулюють роботу інформаційної системи. «Людські програми подібні програмному

15

забезпеченню обладнання» — це звичайна аналогія, яка використовується для ілюстрації ролі програм у системі.

- Люди: кожній системі потрібні люди, щоб функціонувати. Часто елементом системи, який найбільше не помічають, є людина, імовірно, той компонент, який має найбільший вплив на успіх чи невдачу інформаційної системи. До них належать «не тільки користувачі, а й ті, хто керує та обслуговує комп'ютери, ті, хто підтримує дані, і ті, хто підтримує комп'ютерні мережі» [25].
- Зворотній зв'язок: це ще один компонент ІР, який визначає, що ІР може надавати зворотний зв'язок (хоча цей компонент не потрібен для роботи).

### 1.3 Сучасні тенденції веб-розробки

Сучасні інформаційні системи, які використовуються в діяльності агентства знайомств передбачають застосування Веб-додатків, які надають простим користувачам доступ до всіх функцій та послуг агенства.

Веб-додаток (або веб-додаток) — це прикладне програмне забезпечення, яке працює на веб-сервері, на відміну від комп'ютерних програм, які запускаються локально в операційній системі (ОС) пристрою. До веб-додатків користувач отримує доступ через веб-браузер із активним мережевим підключенням. Ці програми програмуються за допомогою змодельованої структури клієнт-сервер — користувачу («клієнту») надаються послуги через сторонній сервер, який розміщений на сторонній стороні. Приклади широко використовуваних веб-додатків включають: веб-пошту, онлайн-роздрібні продажі, онлайн-банкінг та онлайн-аукціони.

Загальна відмінність між динамічною веб-сторінкою будь-якого виду та «веб-додатком» незрозуміла. Веб-сайти, які, швидше за все, називаються "веб-додатками", - це ті, які мають схожі функціональні можливості з програмним забезпеченням для настільних комп'ютерів або мобільним

16

додатком. HTML5 представив явну мовну підтримку для створення програм, які завантажуються як веб-сторінки, але можуть зберігати дані локально та продовжувати працювати в автономному режимі.

Односторінкові програми більше схожі на програми, оскільки вони відкидають більш типову веб-парадигму переміщення між різними сторінками з різними URL-адресами. Це пов'язано з тим, що окремі компоненти можна замінити або оновити без необхідності оновлювати всю веб-сторінку. Односторінкові фреймворки можуть бути використані для прискорення розробки такого веб-додатка для мобільної платформи, оскільки він здатний заощадити пропускну здатність, а також припинити завантаження зовнішніх файлів.

У попередніх моделях обчислень, таких як клієнт-сервер, навантаження на програму розподілялося між кодом на сервері та кодом, встановленим на кожному клієнті локально. Іншими словами, програма мала власну попередньо скомпільовану клієнтську програму, яка слугувала її інтерфейсом користувача і повинна була окремо встановлюватися на персональному комп'ютері кожного користувача.

Оновлення до серверного коду програми зазвичай також вимагає оновлення коду на стороні клієнта, встановленого на кожній робочій станції користувача, що збільшує витрати на підтримку та знижує продуктивність. Крім того, як клієнтський, так і серверний компоненти програми зазвичай були тісно прив'язані до певної комп'ютерної архітектури та операційної системи, і перенесення їх на інші часто було надзвичайно дорогим для всіх додатків, крім найбільших (Сьогодні нативні програми для мобільних пристроїв також обмежені деякими або всіма вищезгаданими питаннями).

На відміну від цього, веб-додатки використовують веб-документи, написані в стандартному форматі, такому як HTML і JavaScript, які підтримуються різними веб-браузерами. Веб-додатки можна розглядати як специфічний варіант клієнт-серверного програмного забезпечення, де

17

клієнтське програмне забезпечення завантажується на клієнтську машину під час відвідування відповідної веб-сторінки, використовуючи стандартні процедури, такі як HTTP. Оновлення веб-програм клієнта можуть відбуватися при кожному відвідуванні веб-сторінки. Під час сеансу веб-браузер інтерпретує та відображає сторінки та діє як універсальний клієнт для будь-якої веб-програми.

На початку Інтернету кожна окрема веб-сторінка доставлялася клієнту як статичний документ, але послідовність сторінок все ще могла забезпечувати інтерактивний досвід, оскільки введення користувача поверталось через елементи веб-форми, вбудовані в розмітку сторінки. Однак кожна істотна зміна веб-сторінки вимагала повернення на сервер, щоб оновити всю сторінку.

У 1995 році Netscape представив мову сценаріїв на стороні клієнта під назвою JavaScript, що дозволяє програмістам додавати деякі динамічні елементи до інтерфейсу користувача, який працював на стороні клієнта.

Таким чином, замість надсилання даних на сервер, щоб створити всю веб-сторінку, вбудовані сценарії завантаженої сторінки можуть виконувати різні завдання, такі як перевірка введених даних або відображення/приховування частин сторінки.[34]

У 1996 році Macromedia представила Flash, програвач векторної анімації, який можна було додати до браузерів як плагін для вбудовування анімації на веб-сторінки. Це дозволило використовувати мову сценаріїв для програмування взаємодії на стороні клієнта без необхідності спілкуватися з сервером.

У 1999 році концепція «веб-додаток» була представлена мовою Java у специфікації сервлетів версії 2.2.[32] На той час і JavaScript, і XML вже були розроблені, але Ajax ще не був придуманий, а об'єкт XMLHttpRequest лише нещодавно був представлений в Internet Explorer 5 як об'єкт ActiveX.

18

У 2005 році був придуманий термін Ajax, і такі програми, як Gmail, почали робити свої клієнтські сторони все більш інтерактивними. Сценарій веб-сторінки може зв'язатися з сервером для зберігання/отримання даних, не завантажуючи всю веб-сторінку.[3]

У 2007 році Стів Джобс оголосив, що веб-додатки, розроблені на HTML5 з використанням архітектури AJAX, будуть стандартним форматом для додатків для iPhone. Набір для розробки програмного забезпечення (SDK) не потрібен, і програми будуть повністю інтегровані в пристрій за допомогою браузера Safari. Пізніше ця модель була переведена на App Store, щоб запобігти джейлбрейкерам і заспокоїти розчарованих розробників.

У 2014 році було завершено розробку HTML5, який надає графічні та мультимедійні можливості без потреби клієнтських плагінів. HTML5 також збагатив семантичний зміст документів. API та об'єктна модель документа (DOM) більше не є запізнілими, а є фундаментальними частинами специфікації HTML5. WebGL API проклав шлях до розширеної 3D-графіки на основі полотна HTML5 і мови JavaScript. Вони мають важливе значення для створення дійсно незалежних від платформи та браузера багатофункціональних веб-додатків.

У 2016 році під час щорічної конференції Google IO Ерік Бідельман (старший інженер з розробки програм) представив прогресивні веб-додатки (PWA) як новий стандарт у веб-розробці. Джефф Бертофт, головний програмний менеджер Microsoft, сказав, що «Google лідирував із прогресивними веб-додатками, і після тривалого процесу ми вирішили, що нам потрібно повністю підтримувати його». Таким чином, і Microsoft, і Google підтримували стандарт PWA.

За допомогою JavaScript, CSS та історично Java, Flash, Silverlight можливі спеціальні методи програми, такі як малювання на екрані, відтворення звуку та доступ до клавіатури та миші. Багато служб працювали, щоб об'єднати все це в більш знайомий інтерфейс, який нагадує зовнішній

19

вигляд операційної системи. Ці технології також підтримують такі методи загального призначення, як перетягування. Веб-розробники часто використовують сценарії на стороні клієнта, щоб додати функціональність, особливо для створення інтерактивного досвіду, який не вимагає перезавантаження сторінки. Нещодавно були розроблені технології для координації сценаріїв на стороні клієнта з технологіями на стороні сервера, такими як ASP.NET, J2EE, Perl/Plack і PHP.

Аjax, техніка веб-розробки, що використовує комбінацію різних технологій, є прикладом технології, яка створює більш інтерактивний досвід. Програми зазвичай розбиваються на логічні блоки, які називаються «рівні», де кожному ярусу призначається роль.[34] Традиційні програми складаються лише з 1 рівня, який знаходиться на клієнтській машині, але веб-додатки за своєю природою піддаються n-рівневому підходу.[34] Хоча можливо багато варіацій, найпоширенішою структурою є трирівнева програма.[34] У своїй найпоширенішій формі три рівні називаються презентацією, додатком і зберіганням у такому порядку.

Веб-браузер — це перший рівень (презентація), механізм, що використовує деякі технології динамічного веб-контенту (наприклад, ASP, CGI, ColdFusion, Dart, JSP/Java, Node.js, PHP, Python або Ruby on Rails) — це середній рівень (логіка програми), а база даних — це третій рівень (сховище).[34] Веб-браузер надсилає запити на середній рівень, який обслуговує їх, виконуючи запити та оновлення бази даних, і створює інтерфейс користувача.

Для більш складних додатків 3-рівневе рішення може виявитися недостатнім, і може бути корисно використовувати n-рівневий підхід, де найбільша перевага полягає в тому, щоб розбити бізнес-логіку, яка знаходиться на рівні програми, на більш детальну модель.[34] Іншою перевагою може бути додавання рівня інтеграції, який відокремлює рівень даних від решти рівнів, забезпечуючи простий у використанні інтерфейс для доступу до даних.[34] Наприклад, доступ до даних клієнта буде

20

здійснюватися шляхом виклику функції "list\_clients()" замість того, щоб робити запит SQL безпосередньо до таблиці клієнта в базі даних. Це дозволяє замінити базову базу даних без будь-яких змін на інших рівнях.[34]

Деякі люди розглядають веб-додаток як дворівневу архітектуру. Це може бути «розумний» клієнт, який виконує всю роботу та запитує «німий» сервер, або «німий» клієнт, який покладається на «розумний» сервер.[34] Клієнт буде обробляти рівень презентації, сервер матиме базу даних (рівень сховища), а бізнес-логіка (рівень програми) буде на одному з них або на обох. [34] Хоча це збільшує масштабованість програм і відокремлює дисплей і базу даних, це все одно не дозволяє справжньої спеціалізації шарів, тому більшість програм переросте цю модель.[34]

Веб-сайт - це сукупність веб-сторінок та пов'язаного вмісту, які ідентифікуються загальним доменним ім'ям та публікуються принаймні на одному веб-сервері. Помітними прикладами є wikipedia.org, google.com та amazon.com.

Усі загальнодоступні веб-сайти в сукупності складають Всесвітню павутину. Існують також приватні веб-сайти, доступ до яких доступний лише в приватній мережі, наприклад, внутрішній веб-сайт компанії для її співробітників.

Веб-сайти, як правило, присвячені певній темі або меті, такі як новини, освіта, комерція, розваги чи соціальні мережі. Гіперпосилання між веб-сторінками спрямовує навігацію по сайту, яке часто починається з домашньої сторінки.

Користувачі можуть отримати доступ до веб-сайтів на різних пристроях, включаючи настільні, ноутбуки, планшети та смартфони. Програма, що використовується на цих пристроях, називається веб-браузером.

Всесвітня павутина (WWW) була створена в 1990 році британським фізиком ЦЕРН Тімом Бернерсом-Лі. [1] 30 квітня 1993 р. ЦЕРН оголосив, що

21

Всесвітня павутина буде безкоштовною для використання будь-ким [2]. До впровадження протоколу передачі гіпертексту (НТТР) для отримання окремих файлів із сервера використовувались інші протоколи, такі як протокол передачі файлів та протокол gopher. Ці протоколи пропонують просту структуру каталогів, в якій користувач переходить і де він обирає файли для завантаження. Документи найчастіше подавалися у вигляді текстових файлів без форматування або кодувались у форматах текстового процесора.

Веб-сайти можуть використовуватися по-різному: персональний веб-сайт, корпоративний веб-сайт компанії, урядовий веб-сайт, веб-сайт організації тощо. Веб-сайти можуть бути роботою приватної особи, бізнесу чи іншої організації і, як правило, присвячені конкретна тема або мета. Будь-який веб-сайт може містити гіперпосилання на будь-який інший веб-сайт, тому різниця між окремими сайтами, як сприймається користувачем, може бути розмитою.

Деякі веб-сайти вимагають реєстрації користувачів або передплати для доступу до вмісту. Приклади веб-сайтів, на які здійснюється підписка, включають багато бізнес-сайтів, веб-сайти новин, веб-сайти академічних журналів, ігрові веб-сайти, веб-сайти спільного використання файлів, дошки оголошень, веб-адреси електронної пошти, веб-сайти соціальних мереж, веб-сайти, що надають дані про фондовий ринок у режимі реального часу, а також веб-сайти, що надають різні інші послуги.

Хоча "веб-сайт" був оригінальним написанням (іноді з великої літери "Веб-сайт", оскільки "Веб" є власним іменником, коли йдеться про Всесвітню павутину), цей варіант став рідкісним, а "веб-сайт" став стандартним написанням. Усі основні керівництва стилем, такі як Чиказький посібник стилю [3] та AP Stylebook [34], відображають цю зміну.

Статичний веб-сайт - це веб-сторінки, які мають веб-сторінки, що зберігаються на сервері у форматі, який надсилається клієнтському веб-

22

браузеру. В основному він кодується мовою розмітки гіпертексту (HTML); Каскадні таблиці стилів (CSS) використовуються для контролю зовнішнього вигляду за базовим HTML. Зображення зазвичай використовуються для отримання бажаного вигляду та як частина основного змісту.

Аудіо чи відео також можна вважати "статичним" вмістом, якщо він відтворюється автоматично або, як правило, є неінтерактивним. Цей тип веб-сайтів зазвичай відображає однакову інформацію для всіх відвідувачів. Подібно до роздачі друкованої брошури клієнтам або клієнтам, статичний веб-сайт, як правило, надає послідовну стандартну інформацію протягом тривалого періоду часу.

Незважаючи на те, що власник веб-сайту може періодично робити оновлення, редагування тексту, фотографій та іншого вмісту здійснюється вручну, і може знадобитися базові навички дизайну веб-сайту та програмне забезпечення. Прості форми або маркетингові приклади веб-сайтів, такі як класичний веб-сайт, веб-сайт із п'ятьма сторінками або веб-сайт брошури, часто є статичними веб-сайтами, оскільки вони представляють користувачеві заздалегідь визначену, статичну інформацію. Це може включати інформацію про компанію та її продукти та послуги через текст, фотографії, анімацію, аудіо / відео та навігаційні меню.

Статичні веб-сайти все ще можуть використовувати серверні компоненти (SSI) як зручність редагування, наприклад, спільний доступ до загального рядка меню на багатьох сторінках. Оскільки поведінка сайту до читача все ще залишається статичним, це не вважається динамічним сайтом.

Динамічний веб-сайт - це веб-сайт, який часто і автоматично змінюється або налаштовується. Динамічні сторінки на сервері генеруються "на льоту" за допомогою комп'ютерного коду, який виробляє HTML (CSS відповідає за зовнішній вигляд і, отже, статичні файли).

Існує широкий спектр програмних систем, таких як CGI, Java-сервлети та Java Server Pages (JSP), Active Server Pages і ColdFusion (CFML), які

23

доступні для створення динамічних веб-систем та динамічних веб-сайтів. Різні фреймворки веб-програм та системи веб-шаблонів доступні для загальноновживаних мов програмування, таких як Perl, PHP, Python та Ruby, щоб полегшити та спростити створення складних динамічних веб-сайтів.

Сайт може відображати поточний стан діалогу між користувачами, відстежувати мінливу ситуацію або надавати інформацію певним чином персоналізовану відповідно до вимог окремого користувача. Наприклад, коли запитується головна сторінка сайту новин, код, що працює на веб-сервері, може поєднувати збережені фрагменти HTML із новинами, отриманими з бази даних або іншого веб-сайту за допомогою RSS, щоб створити сторінку, що включає найсвіжішу інформацію.

Динамічні сайти можуть бути інтерактивними, використовуючи HTML-форми, зберігаючи та зчитуючи файли cookie браузера, або створюючи ряд сторінок, що відображають попередню історію кліків. Інший приклад динамічного вмісту - це коли роздрібний веб-сайт з базою даних медіа-продуктів дозволяє користувачеві вводити запит на пошук, наприклад за ключовим словом "Бітлз". У відповідь вміст веб-сторінки спонтанно змінить те, як він виглядав раніше, а потім відобразить список продуктів "Бітлз", таких як компакт-диски, DVD-диски та книги. Динамічний HTML використовує код JavaScript, щоб вказувати веб-браузеру, як інтерактивно змінювати вміст сторінки. Одним із способів моделювання певного типу динамічного веб-сайту, уникаючи втрати продуктивності ініціювання динамічного механізму для кожного користувача або підключення, є періодична автоматична регенерація великої серії статичних сторінок.

Ранні веб-сайти мали лише текст, а незабаром і зображення. Потім плагіни веб-браузера використовувались для додавання аудіо, відео та інтерактивності (наприклад, для багатofункціональної програми Інтернету, яка відображає складність настільної програми, як текстовий процесор). Прикладами таких плагінів є Microsoft Silverlight, Adobe Flash, Adobe

24

Shockwave та аплети, написані на Java. HTML 5 містить положення щодо аудіо та відео без плагінів. JavaScript також вбудований у більшість сучасних веб-браузерів і дозволяє творцям веб-сайтів надсилати код веб-браузеру, який вказує йому, як інтерактивно змінювати вміст сторінки та спілкуватися з веб-сервером, якщо це необхідно.

Внутрішнє представлення вмісту браузера відомо як об'єктна модель документа (DOM), а техніка - динамічний HTML.

WebGL (Web Graphics Library) - це сучасний API JavaScript для візуалізації інтерактивної 3D-графіки без використання плагінів. Це дозволяє інтерактивний контент, такий як 3D-анімація, візуалізація та відео-пояснення, представленим користувачам найбільш інтуїтивно зрозумілим способом. [5]

Тенденція 2010 року на веб-сайтах під назвою "адаптивний дизайн" забезпечила найкращий досвід перегляду, оскільки надає користувачам макет на основі пристрою. Ці веб-сайти змінюють свій макет відповідно до пристрою або мобільної платформи, що забезпечує багатий досвід користувачів. [6]

Веб-сайти можна розділити на дві великі категорії - статичні та інтерактивні. Інтерактивні сайти є частиною спільноти веб-сайтів Web 2.0 і дозволяють взаємодіяти між власником сайту та відвідувачами або користувачами сайту. Статичні сайти обслуговують або збирають інформацію, але не дозволяють взаємодіяти з аудиторією або користувачами безпосередньо.

Деякі веб-сайти є інформаційними або виготовляються ентузіастами або для особистого користування чи розваги. Багато веб-сайтів мають на меті заробляти гроші, використовуючи одну або кілька бізнес-моделей, зокрема:

- Розміщення цікавого контенту та продаж контекстної реклами або шляхом прямих продажів, або через рекламну мережу.
- Електронна комерція: товари чи послуги купуються безпосередньо через веб-сайт

25

- Реклама товарів або послуг, доступних у цегельному та будівельному бізнесі
- Freemium: базовий вміст доступний безкоштовно, але преміум-вміст вимагає оплати (наприклад, веб-сайт WordPress, це платформа з відкритим кодом для створення блогу чи веб-сайту).

Деякі веб-сайти можуть бути включені в одну або кілька з цих категорій.

Наприклад, веб-сайт бізнесу може рекламувати продукцію бізнесу, але також може розміщувати інформативні документи, такі як технічні документи.

Існують також численні підкатегорії до перерахованих вище. Наприклад, порносайт - це певний тип веб-сайту електронної комерції або бізнес-сайту (тобто він намагається продати членство для доступу до свого сайту) або має можливості соціальних мереж. Фан-сайт може бути посвятою власника певній знаменитості.

Веб-сайти обмежені архітектурними обмеженнями (наприклад, обчислювальна потужність, присвячена веб-сайту). Дуже великі веб-сайти, такі як Facebook, Yahoo !, Microsoft та Google, використовують багато серверів та обладнання для балансування навантажень, наприклад, комутатори служб вмісту Cisco, для розподілу навантажень відвідувачів на декілька комп'ютерів у різних місцях. На початок 2011 року Facebook використовував 9 центрів обробки даних із приблизно 63000 серверами.

У лютому 2009 року компанія Netcraft, компанія з моніторингу Інтернету, яка відслідковувала зростання Інтернету з 1995 р., Повідомила, що в 2009 р. Існувало 215 675 903 веб-сайтів з доменними іменами та вмістом на них, порівняно з лише 19 732 веб-сайтами в серпні 1995 р. [8] Після досягнення 1 мільярда веб-сайтів у вересні 2014 року, етап, підтверджений NetCraft в огляді веб-серверів за жовтень 2014 року, і те, що Інтернет-статистика стала першим, хто повідомив про це, про що свідчить цей твіт від самого винахідника Всесвітньої мережі, Тіма Бернерса Лі - кількість веб-

26

сайтів у світі згодом зменшилась, повернувшись до рівня нижче 1 мільярда. Це пов'язано з щомісячними коливаннями кількості неактивних веб-сайтів. Кількість веб-сайтів до березня 2016 р. Продовжувала зростати до понад 1 млрд. І з тих пір продовжує зростати [9].

Веб-розробка - це робота, пов'язана з розробкою веб-сайту для Інтернету (Всесвітня павутина) або інтрамережі (приватна мережа). [1] Веб-розробка може варіюватися від розробки простої однієї статичної сторінки простого тексту до складних Інтернет-програм (Інтернет-додатків), електронного бізнесу та послуг соціальних мереж. Більш повний перелік завдань, до яких зазвичай відноситься веб-розробка, може включати веб-інженерію, веб-дизайн, розробку веб-вмісту, зв'язок з клієнтом, сценарії на стороні клієнта / сервера, налаштування безпеки веб-сервера та мережі та розвиток електронної комерції.

Серед веб-спеціалістів "веб-розробка" зазвичай відноситься до основних недизайнерських аспектів побудови веб-сайтів: написання розмітки та кодування. [32] Веб-розробка може використовувати системи управління вмістом (CMS), щоб зробити зміст вмістом простішим та доступнішим з базовими технічними навичками.

Для великих організацій та бізнесу групи веб-розробників можуть складатися з сотень людей (веб-розробників) і дотримуватися стандартних методів, таких як Agile, під час розробки веб-сайтів. Менші організації можуть вимагати лише одного постійного або підрядного розробника, або вторинне призначення на відповідні посади, такі як графічний дизайнер або технік інформаційних систем. Веб-розробка може бути спільним зусиллям між департаментами, а не областю призначеного департаменту.

Існує три різновиди спеціалізації веб-розробників: розробник з інтерфейсу, розробник з інтерфейсу та розробник із повним стеком. Інтернетні розробники несуть відповідальність за поведінку та візуальні

27

ефекти, які працюють у браузері користувача, тоді як інтерфейсні розробники мають справу з серверами.

З моменту комерціалізації Інтернету веб-розробка стала зростаючою галуззю. Зростання цієї галузі відбувається за рахунок підприємств, які бажають використовувати свій веб-сайт для реклами та продажу товарів та послуг споживачам. [3]

Існує багато інструментів з відкритим кодом для веб-розробки, таких як BerkeleyDB, GlassFish, стек LAMP (Linux, Apache, MySQL, PHP) та Perl / Plack. Це дозволило звести витрати на навчання веб-розробці до мінімуму.

Ще одним фактором, що сприяє зростанню галузі, стало зростання простого у використанні програмного забезпечення для веб-розробки WYSIWYG, такого як Adobe Dreamweaver, BlueGriffon та Microsoft Visual Studio. Для використання такого програмного забезпечення все ще необхідні знання мови розмітки HyperText (HTML) або мов програмування, але основи можна швидко вивчити та впровадити.

Постійно зростаючий набір інструментів та технологій допоміг розробникам створювати більш динамічні та інтерактивні веб-сайти. Крім того, веб-розробники тепер допомагають надавати програми як веб-служби, які традиційно були доступні лише як програми на настільному комп'ютері. Це дало багато можливостей для децентралізації розповсюдження інформації та засобів масової інформації.

Приклади можна побачити із зростанням хмарних сервісів, таких як Adobe Creative Cloud, Dropbox та Google Drive. Ці веб-служби дозволяють користувачам взаємодіяти з програмами з багатьох місць, замість того, щоб бути прив'язаними до певної робочої станції для свого середовища програм.

Прикладами кардинальних перетворень у спілкуванні та комерції на чолі з веб-розробкою є електронна комерція. Інтернет-сайти аукціонів, такі як eBay, змінили спосіб споживачів знаходити та купувати товари та послуги. Інтернет-магазини, такі як Amazon.com та Buy.com (серед багатьох інших),

28

змінити досвід покупок та торгівлі для багатьох споживачів. Іншим прикладом трансформаційного спілкування, керованого веб-розробкою, є блог.

Веб-програми, такі як WordPress та Movable Type, створили середовища для ведення блогів для окремих веб-сайтів. Поширене використання систем управління вмістом з відкритим кодом та систем управління корпоративним контентом розширило вплив веб-розробки на онлайн-взаємодію та спілкування.

Розробка веб-сайтів також вплинула на особисті мережі та маркетинг. Веб-сайти вже не є просто інструментами для роботи чи комерції, а ширше служать для спілкування та соціальних мереж. Такі веб-сайти, як Facebook та Twitter, надають користувачам платформу для спілкування, а організації - більш особистий та інтерактивний спосіб залучення громадськості.

Веб-розробка враховує багато міркувань безпеки, таких як перевірка помилок при введенні даних через форми, фільтрація вихідних даних та шифрування. Шкідливі практики, такі як введення SQL, можуть виконуватися користувачами з ненавмисними намірами, але лише з примітивними знаннями про веб-розробку в цілому.

Сценарії можна використовувати для використання веб-сайтів, надаючи несанкціонований доступ зловмисним користувачам, які намагаються збирати таку інформацію, як адреси електронної пошти, паролі та захищений вміст, наприклад номери кредитних карток.

Дещо з цього залежить від серверного середовища, на якому запущена мова сценаріїв, наприклад ASP, JSP, PHP, Python, Perl або Ruby, і тому не обов'язково підтримувати самого розробника. Однак рекомендується жорстке тестування веб-додатків перед публічним випуском, щоб запобігти подібним подвигам. Якщо на веб-сайті є якась контактна форма, вона повинна включати в неї поле captcha, яке не дозволяє комп'ютерним програмам автоматично заповнювати форми, а також розсилати пошту.

29

Захист веб-сервера від вторгнень часто називають зміцненням порту сервера. Багато технологій застосовуються для захисту інформації в Інтернеті, коли вона передається з одного місця в інше. Наприклад, сертифікати TLS (або "сертифікати SSL") видаються органами сертифікації для запобігання шахрайству в Інтернеті. Багато розробників часто використовують різні форми шифрування при передачі та зберіганні конфіденційної інформації. Базове розуміння питань безпеки інформаційних технологій часто є частиною знань веб-розробника.

Оскільки нові веб-програми виявляють нові діри в безпеці навіть після тестування та запуску, оновлення виправлень безпеки часто трапляються для широко використовуваних програм. Часто робота веб-розробників - постійно оновлювати програми, коли випускаються виправлення безпеки та виявляються нові проблеми безпеки.

## 1.4 Аналіз існуючих рішень

Найбільш відомим аналогом, який можна кастомізувати під агентство знайомств — 1С:Підприємство.

1С:Підприємство — це платформа розробки, розроблена компанією 1С для швидкого створення програмного забезпечення для автоматизації бізнесу, що легко налаштовується.

1С:Підприємство — це повноцінна платформа з низьким кодом, яка надає готову до використання інфраструктуру та інструменти для швидкої розробки бізнес-додатків, таких як ERP, POS, WMS або інше корпоративне програмне забезпечення на замовлення. «1С:Підприємство» використовує доменний підхід до розробки програмного забезпечення для бізнесу. Розробка додатків у рамках 1С:Підприємства здійснюється за допомогою високорівневої об'єктно-орієнтованої мови в концептуальній моделі, що точно наближає бізнес-завдання.[32]

30

Візуальне редагування використовується протягом усього процесу розробки «ІС:Підприємство». Компанія ІС стверджує, що це зводить обсяг програмування як такого до мінімуму, відповідно до парадигми розробки низького коду. Платформа ІС:Підприємство складається з таких частин:

- Рівень даних/процесів, який надає системні функції, такі як підтримка різноманітних систем керування базами даних, сервер додатків, компоненти веб-сервера та компоненти веб-сервісів.
- Бізнес-компоненти – попередньо визначені, візуально настроювані будівельні блоки (шаблони), які широко використовуються в корпоративних додатках – каталоги, документи, книги, процеси розрахунків та настроювана логіка бізнес-процесів.

## 1.5 Постановка задачі

Основна задача полягає в створенні власної реалізації системи управління діяльністю агентства знайомств.

В ході реалізації має бути забезпечена автоматизація виконання наступних функцій:

- реєстрація користувачів;
- авторизація користувачів;
- перегляд анкет чоловіків;
- додавання анкет чоловіків;
- перегляд анкет жінок;
- додавання анкет жінок;
- перегляд сформованих пар;
- додавання нових пар.

## РОЗДІЛ 2

### АНАЛІЗ ІНСТРУМЕНТАЛЬНИХ ЗАСОБІВ РЕАЛІЗАЦІЇ

#### 2.1 Огляд мови програмування

Java — це високорівнева, заснована на класах, об'єктно-орієнтована мова програмування, яка розроблена так, щоб мати якомога менше залежностей реалізації. Це мова програмування загального призначення, призначена для того, щоб програмісти могли писати один раз, запускати в будь-якому місці (WORA),[26] що означає, що скомпільований код Java може працювати на всіх платформах, які підтримують Java, без необхідності перекомпіляції.[27]

Програми Java зазвичай компілюються у байт-код, який може працювати на будь-якій віртуальній машині Java (JVM) незалежно від базової архітектури комп'ютера. Синтаксис Java подібний до C і C++, але має менше засобів низького рівня, ніж будь-який з них. Середовище виконання Java надає динамічні можливості (такі як відображення та модифікація коду під час виконання), які зазвичай недоступні в традиційних скомпільованих мовах. Станом на 2019 рік Java була однією з найпопулярніших мов програмування, що використовуються відповідно до GitHub,[30] особливо для веб-додатків клієнт-сервер, із 9 мільйонами розробників.[31]

Спочатку Java була розроблена Джеймсом Гослінгом у Sun Microsystems і випущена в травні 1995 року як основний компонент Java-платформи Sun Microsystems. Оригінальні та довідкові компілятори Java, віртуальні машини та бібліотеки класів спочатку були випущені Sun за власними ліцензіями. Станом на травень 2007 року, відповідно до специфікацій Java Community Process, Sun переліцензувала більшість своїх технологій Java за ліцензією лише GPL-2.0. Oracle пропонує власну віртуальну машину HotSpot Java, однак офіційною довідковою реалізацією є

32

OpenJDK JVM, яка є безкоштовним програмним забезпеченням з відкритим вихідним кодом і використовується більшістю розробників і є JVM за замовчуванням для майже всіх дистрибутивів Linux.

Станом на березень 2022 року Java 18 є останньою версією, тоді як Java 17, 11 і 8 є поточними версіями довгострокової підтримки (LTS). Oracle випустила останнє загальнодоступне оновлення за нульовою вартістю для застарілої версії Java 8 LTS у січні 2019 року для комерційного використання, хоча в іншому випадку вона все ще підтримуватиме Java 8 з загальнодоступними оновленнями для особистого використання на невизначений термін. Інші виробники почали пропонувати безкоштовні збірки OpenJDK 8 і 11, які все ще отримують безпеку та інші оновлення. Oracle (та інші) настійно рекомендують видаляти застарілі та не підтримувані версії Java через невирішені проблеми безпеки в старих версіях.[22] Oracle радить своїм користувачам негайно перейти на підтримувану версію, наприклад одну з версій LTS (8, 11, 17).

Однією з цілей дизайну Java є переносимість, що означає, що програми, написані для платформи Java, повинні працювати подібним чином на будь-якій комбінації апаратного забезпечення та операційної системи з належною підтримкою часу виконання. Це досягається шляхом компіляції коду мови Java в проміжне представлення, яке називається байт-кодом Java, замість безпосереднього машинного коду, специфічного для архітектури.

Інструкції байт-коду Java аналогічні машинному коду, але вони призначені для виконання віртуальною машиною (VM), написаною спеціально для апаратного забезпечення хоста. Кінцеві користувачі зазвичай використовують Java Runtime Environment (JRE), встановлену на своєму пристрої для автономних програм Java або веб-браузер для Java-апплетів. Стандартні бібліотеки надають загальний спосіб доступу до специфічних для хоста функцій, таких як графіка, потоки та мережа.

33

Використання універсального байт-коду спрощує перенесення. Однак накладні витрати на інтерпретацію байт-коду в машинні інструкції змушували інтерпретовані програми майже завжди працювати повільніше, ніж рідні виконувані файли. Компілятори Just-in-time (JIT), які компілюють байт-код у машинний код під час виконання, були представлені на ранньому етапі.

Компілятор Hotspot Java насправді є двома компіляторами в одному; і з GraalVM (включений, наприклад, у Java 11, але вилучений з Java 16), що дозволяє багаторівневу компіляцію.[31] Сама Java не залежить від платформи і адаптована до конкретної платформи, на якій вона має працювати, за допомогою віртуальної машини Java (JVM), яка перекладає байт-код Java на машинну мову платформи.[34]

Програми, написані на Java, мають репутацію повільніших і вимагають більше пам'яті, ніж програми, написані на C++.[33] Однак швидкість виконання програм Java значно покращилася із запровадженням компіляції «точно вчасно» в 1997/1998 роках для програм на Java. Java 1.1.[33], додавання мовних функцій, які підтримують кращий аналіз коду (наприклад, внутрішні класи, клас StringBuilder, необов'язкові твердження тощо), а також оптимізації у віртуальній машині Java, наприклад HotSpot, ставши JVM Sun за замовчуванням у 2000 році. З Java 1.5 продуктивність була покращена за допомогою додавання пакета java.util.concurrent, включаючи безблоковані реалізації ConcurrentHashMap та інших багатоядерних колекцій, а також покращено з Java 1.6.

Java використовує автоматичний збірник сміття для управління пам'яттю в життєвому циклі об'єкта. Програміст визначає час створення об'єктів, а середовище виконання Java відповідає за відновлення пам'яті, коли об'єкти більше не використовуються.

Як тільки посилання на об'єкт не залишаються, недоступна пам'ять стає придатною для автоматичного звільнення збирачем сміття. Щось подібне

34

до витоку пам'яті все ще може статися, якщо код програміста містить посилання на об'єкт, який більше не потрібен, як правило, коли об'єкти, які більше не потрібні, зберігаються в контейнерах, які все ще використовуються. Якщо викликаються методи для неіснуючого об'єкта, створюється виняток нульового покажчика.[37]

Одна з ідей, що лежить в основі моделі автоматичного управління пам'яттю Java, полягає в тому, що програмісти можуть позбутися від тягара виконання ручного керування пам'яттю. У деяких мовах пам'ять для створення об'єктів неявно виділяється в стеку або явно виділяється і звільняється з купи. В останньому випадку відповідальність за управління пам'яттю покладається на програміста. Якщо програма не звільняє об'єкт, відбувається витік пам'яті.

Якщо програма намагається отримати доступ або звільнити пам'ять, яка вже була вивільнена, результат буде невизначеним і важко передбачити, і програма, швидше за все, стане нестабільною або аварійно завершить роботу.

Це можна частково виправити за допомогою розумних покажчиків, але вони додають накладних витрат і ускладнюють. Зауважте, що збір сміття не запобігає витоку логічної пам'яті, тобто тих, де пам'ять все ще посилається, але ніколи не використовується.

Збір сміття може статися в будь-який момент. В ідеалі це відбуватиметься, коли програма неактивна. Він гарантовано спрацьовує, якщо в купі недостатньо вільної пам'яті для виділення нового об'єкта; це може призвести до миттєвої зупинки програми. Явне керування пам'яттю неможливе в Java.

Java не підтримує арифметику вказівників у стилі C/C++, де адресами об'єктів можна арифметично маніпулювати (наприклад, додаючи або віднімаючи зміщення). Це дозволяє збірнику сміття переміщувати об'єкти, на які посилаються, і гарантує безпеку та безпеку типу.

35

Як і в C++ та деяких інших об'єктно-орієнтованих мовах, змінні примітивних типів даних Java зберігаються або безпосередньо в полях (для об'єктів), або в стеку (для методів), а не в купі, як це зазвичай вірно для непримітивних даних. типи (але див. escape-аналіз). Це було свідоме рішення дизайнерів Java з міркувань продуктивності.

Java містить кілька типів збирачів сміття. Починаючи з Java 9, HotSpot використовує Garbage First Garbage Collector (G1GC) за замовчуванням.[37]

Однак є також кілька інших збирачів сміття, які можна використовувати для керування купою. Для більшості програм на Java достатньо G1GC. Раніше в Java 8 використовувався Parallel Garbage Collector.

Вирішення проблеми управління пам'яттю не звільняє програміста від тягаря належної обробки інших видів ресурсів, таких як мережеві з'єднання або підключення до бази даних, дескриптори файлів тощо, особливо за наявності **ВИНЯТКІВ.**

Jakarta Servlet (раніше Java Servlet) — це програмний компонент Java, який розширює можливості сервера. Хоча сервлети можуть відповідати на багато типів запитів, вони найчастіше реалізують веб-контейнери для розміщення веб-додатків на веб-серверах і, таким чином, кваліфікуються як веб-API сервлетів на стороні сервера. Такі веб-сервлети є аналогом Java для інших технологій динамічного веб-контенту, таких як PHP і ASP.NET.

Загальна характеристика Сервлет Jakarta

Сервлет Jakarta обробляє або зберігає клас Java в Jakarta EE, який відповідає API сервлетів Jakarta [37], стандарту для реалізації класів Java, які відповідають на запити. Сервлети в принципі можуть спілкуватися за будь-яким протоколом клієнт-сервер, але найчастіше вони використовуються з HTTP. Таким чином, «сервлет» часто використовується як скорочення «сервлет HTTP».[32]

Таким чином, розробник програмного забезпечення може використовувати сервлет для додавання динамічного вмісту на веб-сервер за

36

допомогою платформи Java. Згенерований зміст зазвичай являє собою HTML, але можуть бути й інші дані, такі як XML і частіше JSON. Сервлети можуть підтримувати стан змінних сеансу в багатьох транзакціях сервера за допомогою файлів cookie HTTP або зіставлення URL-адрес.

API Jakarta Servlet певною мірою було замінено двома стандартними технологіями Java для веб-сервісів:

- веб-сервіси Jakarta RESTful Web Services (JAX-RS 2.0), корисні для служб AJAX, JSON і REST, а також
- Веб-служби Джакарти XML (JAX-WS), корисні для веб-служб SOAP.

Для розгортання та запуску сервлета необхідно використовувати веб-контейнер. Веб-контейнер (також відомий як контейнер сервлетів) по суті є компонентом веб-сервера, який взаємодіє із сервлетами. Веб-контейнер відповідає за керування життєвим циклом сервлетів, зіставлення URL-адреси з певним сервлетом і забезпечення того, що запитувач URL-адрес має правильні права доступу.

API сервлетів, що міститься в ієрархії пакетів Java javax.servlet, визначає очікувані взаємодії веб-контейнера та сервлета.[32]

Сервлет - це об'єкт, який отримує запит і генерує відповідь на основі цього запиту. Базовий пакет сервлетів визначає об'єкти Java для представлення запитів і відповідей сервлетів, а також об'єкти, що відображають параметри конфігурації сервлета та середовище виконання. Пакет javax.servlet.http визначає HTTP-специфічні підкласи загальних елементів сервлетів, включаючи об'єкти керування сеансом, які відстежують численні запити та відповіді між веб-сервером і клієнтом. Сервлети можуть бути запаковані у файл WAR як веб-додаток.

Сервлети можуть бути автоматично створені з Jakarta Server Pages (JSP) компілятором Jakarta Server Pages. Різниця між сервлетами і JSP полягає в тому, що сервлети зазвичай вбудовують HTML в код Java, тоді як JSP вбудовують код Java в HTML. Хоча пряме використання сервлетів для

37

генерації HTML (як показано в прикладі нижче) стало рідкістю, веб-фреймворк MVC вищого рівня в Jakarta EE (JSF) все ще явно використовує технологію сервлетів для низькорівневої обробки запитів/відповідей через FacesServlet. Дещо давнішим є використання сервлетів у поєднанні з JSP у шаблоні, який називається «Модель 2», який є різновидом контролера model-view-controller.

Jakarta Server Pages (JSP; раніше JavaServer Pages) — це набір технологій, які допомагають розробникам програмного забезпечення створювати динамічно створювані веб-сторінки на основі HTML, XML, SOAP або інших типів документів. Випущений у 1999 році компанією Sun Microsystems[1], JSP подібний до PHP і ASP, але використовує мову програмування Java.

Для розгортання та запуску Jakarta Server Pages потрібен сумісний веб-сервер із контейнером сервлетів, наприклад Apache Tomcat або Jetty.

Архітектурно JSP можна розглядати як високорівневу абстракцію сервлетів Java. JSP транслюються в сервлети під час виконання, тому JSP є сервлетом; кожен сервлет JSP кешується і повторно використовується, доки оригінальний JSP не буде змінено.[32]

Серверні сторінки Jakarta можна використовувати окремо або як компонент перегляду серверної моделі-вигляду-контролера, як правило, з JavaBeans як моделлю і сервлетами Java (або структурою, як-от Apache Struts) як контролером. Це тип архітектури Model 2.[37]

JSP дозволяє перемежовувати код Java і певні заздалегідь визначені дії зі статичним вмістом веб-розмітки, таким як HTML. Отримана сторінка компілюється і виконується на сервері для доставки документа.

Скомпільовані сторінки, а також будь-які залежні бібліотеки Java містять байт-код Java, а не машинний код. Як і будь-яка інша програма .jar або Java, код має виконуватися на віртуальній машині Java (JVM), яка

38

взаємодіє з операційною системою хоста сервера, щоб забезпечити абстрактне, нейтральне для платформи середовище.

JSP зазвичай використовуються для доставки документів HTML і XML, але за допомогою OutputStream вони також можуть доставляти інші типи даних.[34]

Веб-контейнер створює неявні об'єкти JSP, такі як запит, відповідь, сесія, програма, конфігурація, сторінка, pageContext, вихід і виняток. JSP Engine створює ці об'єкти на етапі перекладу.

Компілятор сторінок JavaServer — це програма, яка аналізує JSP і перетворює їх у виконувані сервлети Java. Програма такого типу зазвичай вбудовується в сервер додатків і запускається автоматично під час першого доступу до JSP, але сторінки також можуть бути попередньо скомпільовані для кращої продуктивності або скомпільовані як частина процесу збірки для перевірки на наявність помилок.[36]

Деякі контейнери JSP підтримують налаштування того, як часто контейнер перевіряє часові позначки файлу JSP, щоб побачити, чи змінилася сторінка. Зазвичай ця позначка часу встановлюється на короткий інтервал (можливо, секунди) під час розробки програмного забезпечення та довший інтервал (можливо, хвилини або навіть ніколи) для розгорнутого веб-додатка.[37]

## 2.2 Огляд середовища розробки

IntelliJ IDEA — це інтегроване середовище розробки (IDE), написане на Java для розробки комп'ютерного програмного забезпечення. Він розроблений компанією JetBrains (раніше відомий як IntelliJ) і доступний у вигляді ліцензованої спільноти Apache 2[32] та у приватній комерційній версії. Обидва можна використовувати для комерційного розвитку.[35]

39

Перша версія IntelliJ IDEA була випущена в січні 2001 року і була однією з перших доступних середовищ IDE Java з інтегрованими можливостями розширеної навігації та рефакторингу коду.[35].

У звіті InfoWorld за 2010 рік IntelliJ отримав найвищу оцінку центру тестування з чотирьох найкращих інструментів програмування Java: Eclipse, IntelliJ IDEA, NetBeans і JDeveloper.[34]

У грудні 2014 року Google оголосила про версію 1.0 Android Studio, IDE з відкритим кодом для програм Android, засновану на версії IntelliJ IDEA для спільноти з відкритим кодом.[8] Інші середовища розробки, засновані на фреймворку IntelliJ, включають AppCode, CLion, DataGrip, GoLand, PhpStorm, PyCharm, Rider, RubyMine, WebStorm і MPS.[9]

IDE надає певні функції[32], як-от завершення коду шляхом аналізу контексту, навігацію по коду, яка дозволяє безпосередньо переходити до класу або оголошення в коді, рефакторинг коду, налагодження коду[33], літинг і параметри для виправлення невідповідностей за допомогою пропозицій.

IDE забезпечує[32] інтеграцію з інструментами збірки/пакування, такими як grunt, bower, gradle і SBT. Він підтримує системи контролю версій, такі як Git, Mercurial, Perforce і SVN. До таких баз даних, як Microsoft SQL Server, Oracle, PostgreSQL, SQLite і MySQL, можна отримати доступ безпосередньо з IDE у версії Ultimate через вбудовану версію DataGrip.

IntelliJ підтримує плагіни, за допомогою яких можна додати додаткові функції до IDE. Плагіни можна завантажити та встановити або з веб-сайту сховища плагінів IntelliJ, або через вбудовану функцію пошуку та встановлення плагінів IDE. Кожне видання має окремі сховища плагінів, і станом на 2019 рік у виданнях Community та Ultimate є понад 3000 плагінів кожен.[34]

40

### 2.3 Огляд додаткового інструментарію

Мова розмітки гіпертексту або HTML є стандартною мовою розмітки для документів, призначених для відображення у веб-браузері. Цьому можуть допомогти такі технології, як каскадні таблиці стилів (CSS) і мови сценаріїв, такі як JavaScript.

Веб-браузери отримують документи HTML з веб-сервера або з локального сховища і перетворюють документи на мультимедійні веб-сторінки. HTML описує структуру веб-сторінки семантично та оригінально включав підказки для зовнішнього вигляду документа.

Елементи HTML є будівельними блоками сторінок HTML. За допомогою конструкцій HTML зображення та інші об'єкти, такі як інтерактивні форми, можуть бути вбудовані в відтворену сторінку. HTML надає засоби для створення структурованих документів, позначаючи структурну семантику тексту, наприклад заголовки, абзаци, списки, посилання, цитати та інші елементи. Елементи HTML розмежовуються тегами, записаними за допомогою кутових дужок. Такі теги, як `<img />` і `<input />` безпосередньо вносять вміст на сторінку. Інші теги, такі як `<p>`, оточують і надають інформацію про текст документа, а також можуть включати інші теги як піделементи. Браузери не відображають теги HTML, а використовують їх для інтерпретації вмісту сторінки.

HTML може вбудовувати програми, написані мовою сценаріїв, наприклад JavaScript, що впливає на поведінку та вміст веб-сторінок. Включення CSS визначає вигляд і макет вмісту. Консорціум World Wide Web Consortium (W3C), колишній розпорядник HTML і нинішній розпорядник стандартів CSS, заохочує використання CSS замість явного презентаційного HTML з 1997 року.[32] Форма HTML, відома як HTML5, використовується для відображення відео та аудіо, переважно за допомогою елемента `<canvas>` у співпраці з javascript.

41

Розмітка HTML складається з кількох ключових компонентів, включаючи ті, які називаються тегами (та їх атрибутами), типи даних на основі символів, посилання на символи та посилання на сутність. HTML-теги найчастіше зустрічаються парами, як-от `<h1>` і `</h1>`, хоча деякі представляють порожні елементи і тому є непарними, наприклад `<img>`. Першим тегом у такій парі є початковий тег, а другий — кінцевий (їх ще називають відкриваючими та закриваючими тегами).

Іншим важливим компонентом є оголошення типу документа HTML, яке запускає відтворення стандартного режиму.

Нижче наведено приклад класичного "Hello, World!" програма:

```
<!DOCTYPE html>
<html>
  <голова>
    <title>Це назва</title>
  </head>
  <тіло>
    <div>
      <p>Привіт, світ!</p>
    </div>
  </body>
</html>
```

Текст між `<html>` і `</html>` описує веб-сторінку, а текст між `<body>` і `</body>` є видимим вмістом сторінки. Текст розмітки `<title>Це заголовок</title>` визначає заголовок сторінки браузера, що відображається на вкладках і заголовках вікон браузера, а тег `<div>` визначає поділ сторінки, який використовується для легкого оформлення. Між `<head>` і `</head>` для визначення метаданих веб-сторінки можна використовувати елемент `<meta>`.

42

Декларація типу документа `<!DOCTYPE html>` призначена для HTML5. Якщо оголошення не включено, різні браузері повертатимуться до «режиму химерності» для відтворення.[38]

Документи HTML мають на увазі структуру вкладених елементів HTML. Вони вказуються в документі тегами HTML, укладеними в кутові дужки таким чином: `<p>`. [38]

У простому, загальному випадку, екстент елемента вказується парою тегів: «початковий тег» `<p>` і «кінцевий тег» `</p>`. Текстовий вміст елемента, якщо такий є, розміщується між цими тегами.

Теги також можуть містити додаткову розмітку тегів між початком і кінцем, включаючи суміш тегів і тексту. Це вказує на подальші (вкладені) елементи як дочірні елементи батьківського елемента.

Початковий тег також може включати атрибути елемента всередині тегу. Вони вказують іншу інформацію, таку як ідентифікатори розділів у документі, ідентифікатори, які використовуються для прив'язки інформації про стиль до презентації документа, а для деяких тегів, таких як `<img>`, що використовуються для вбудовування зображень, посилання на ресурс зображення в такий формат: ``

Деякі елементи, такі як розрив рядка `<br>` або `<br />`, не дозволяють вбудовувати будь-який вміст, текст або інші теги. Для них потрібен лише один порожній тег (як початковий тег) і не використовується кінцевий тег.

Багато тегів, зокрема закриваючий кінцевий тег для дуже часто використовуваного елемента абзацу `<p>`, є необов'язковими. Браузер HTML або інший агент може зробити висновок про закриття кінця елемента з контексту та структурних правил, визначених стандартом HTML. Ці правила складні та не зрозумілі більшості програмістів HTML.

Таким чином, загальна форма елемента HTML: `<tag attribute1="value1" attribute2="value2">"content"</tag>`. Деякі елементи HTML визначаються як порожні елементи і мають форму `<tag attribute1="value1" attribute2="value2">`.

43

Порожні елементи можуть не містити вмісту, наприклад, тег `<br>` або вбудований тег `<img>`. Ім'я елемента HTML - це ім'я, яке використовується в тегах. Зверніть увагу, що назві кінцевого тегу передує символ косої риски /, і що в порожніх елементах кінцевий тег не є ані обов'язковим, ані дозволеним. Якщо атрибути не згадуються, у кожному випадку використовуються значення за замовчуванням.

Структурна розмітка вказує на призначення тексту.

Наприклад, `<h2>Golf</h2>` встановлює "Гольф" як заголовок другого рівня. Структурна розмітка не позначає жодного конкретного візуалізації, але більшість веб-браузерів мають стилі за замовчуванням для форматування елементів. Вміст можна додатково стилізувати за допомогою каскадних таблиць стилів (CSS).[39]

Презентаційна розмітка вказує на зовнішній вигляд тексту, незалежно від його призначення.

Наприклад, `<b>жирний текст</b>` вказує на те, що пристрої візуального виводу мають відображати жирний текст, але дає мало вказівок, які пристрої, які не в змозі це зробити (наприклад, звукові пристрої, які читають текст вголос), повинні робити. У випадку як `<b>жирного тексту</b>`, так і `<i>курсивного тексту</i>`, існують інші елементи, які можуть мати еквівалентне візуальне відображення, але більш семантичний характер, наприклад `<strong>сильний текст</strong>` і `<em>підкреслений текст</em>` відповідно.

Легше зрозуміти, як слуховий користувальницький агент повинен інтерпретувати останні два елементи. Однак вони не є еквівалентними своїм презентаційним аналогам: було б небажано, щоб програма читання з екрана, наприклад, підкреслювала назву книги, але на екрані таку назву було б виділено курсивом. Більшість елементів розмітки презентації застаріли відповідно до специфікації HTML 4.0 на користь використання CSS для стилізації.

44

Гіпертекстова розмітка перетворює частини документа на посилання на інші документи.

Елемент прив'язки створює гіперпосилання в документі, а його атрибут `href` встановлює цільову URL-адресу посилання. Наприклад, HTML-розмітка `<a href="https://www.google.com/">Вікіпедія</a>` відобразить слово "Вікіпедія" як гіперпосилання. Щоб відобразити зображення як гіперпосилання, елемент `img` вставляється як вміст в елемент `a`. Як і `br`, `img` є порожнім елементом з атрибутами, але без вмісту чи закриваючого тега. `<a href="https://example.org"></a>`.

Більшість атрибутів елемента – це пари ім'я-значення, розділені знаком `=` і записані в початковий тег елемента після імені елемента. Значення можна брати в одинарні або подвійні лапки, хоча значення, що складаються з певних символів, можна залишити без лапок у HTML (але не в XHTML).[38][39] Залишати значення атрибутів без лапок вважається небезпечним.[39] На відміну від атрибутів пари ім'я-значення, є деякі атрибути, які впливають на елемент просто своєю присутністю в початковому тегу елемента[36], як-от атрибут `ismap` для елемента `img`.

Існує кілька загальних атрибутів, які можуть з'являтися в багатьох елементах:

- Атрибут `id` забезпечує унікальний ідентифікатор елемента для всього документа. Це використовується для ідентифікації елемента, щоб таблиці стилів могли змінювати його презентаційні властивості, а сценарії могли змінювати, анімувати або видаляти його вміст або презентацію. Доданий до URL-адреси сторінки, він надає глобальний унікальний ідентифікатор для елемента, як правило, підрозділу сторінки. Наприклад, ідентифікатор «Атрибути» в <https://en.wikipedia.org/wiki/HTML#Attributes>.

45

- Атрибут class забезпечує спосіб класифікації подібних елементів. Це можна використовувати для семантичних або презентаційних цілей. Наприклад, документ HTML може семантично використовувати позначення `<class="notation">`, щоб вказати, що всі елементи з цим значенням класу підпорядковані основному тексту документа. У презентації такі елементи можуть бути зібрані разом і представлені у вигляді виносок на сторінці замість того, щоб з'являтися там, де вони зустрічаються у джерелі HTML. Атрибути класу використовуються семантично в мікроформатах. Можна вказати кілька значень класів; наприклад, `<class="notation important">` поміщає елемент як до нотації, так і до важливих класів.
- Автор може використовувати атрибут style для призначення властивостей презентації певному елементу. Вважається кращою практикою використовувати ідентифікатор елемента або атрибути класу для вибору елемента з таблиці стилів, хоча іноді це може бути занадто громіздким для простого, конкретного або спеціального стилю.
- Атрибут title використовується для додавання підтекстового пояснення до елемента. У більшості браузерів цей атрибут відображається як підказка.
- Атрибут lang визначає природну мову вмісту елемента, яка може відрізнятися від мови решти документа. Наприклад, в англійськом документі:

`<p>Ну що ж, <span lang="fr">c'est la vie</span>, як кажуть у Франції.</p>`

Семантичний HTML – це спосіб написання HTML, який підкреслює значення закодованої інформації над її представленням (виглядом). HTML включав семантичну розмітку з самого початку,[40] але також включав презентаційну розмітку, таку як теги `<font>`, `<i>` і `<center>`. Існують також семантично нейтральні теги `span` і `div`. З кінця 1990-х років, коли каскадні

46

таблиці стилів почали працювати в більшості браузерів, веб-авторів заохочували уникати використання презентаційної HTML-розмітки з метою розділення презентації та вмісту.[40]

Під час дискусії 2001 року про семантичну мережу Тім Бернерс-Лі та інші навели приклади того, як інтелектуальні програмні «агенти» можуть одного дня автоматично сканувати Інтернет і знаходити, фільтрувати та співвідносити раніше не пов'язані опубліковані факти на користь користувачів. [39] Такі агенти навіть зараз не є звичайним явищем, але деякі ідеї Web 2.0, мішапи та веб-сайти порівняння цін можуть наблизитися. Основна відмінність між цими гібридами веб-додатків і семантичними агентами Бернерса-Лі полягає в тому, що поточну агрегацію та гібридизацію інформації зазвичай розробляють веб-розробники, які вже знають веб-розташування та семантику API конкретних даних, які вони бажають. м'яти, порівнювати та з'єднувати.

Важливим типом веб-агента, який сканує та читає веб-сторінки автоматично, не знаючи, що він може знайти, є веб-сканер або пошукова система. Ці програмні агенти залежать від семантичної чіткості веб-сторінок, які вони знаходять, оскільки вони використовують різні методи та алгоритми для читання та індексації мільйонів веб-сторінок на день і надають користувачам Інтернету засоби пошуку, без яких корисність Всесвітньої мережі була б значно зменшена.

Щоб павуки пошукових систем могли оцінити важливість фрагментів тексту, які вони знаходять у документах HTML, а також для тих, хто створює мешапи та інші гібриди, а також для більш автоматизованих агентів у міру їх розробки, наявні семантичні структури у HTML необхідно широко і рівномірно застосовувати, щоб виявити значення опублікованого тексту.[40] Теги презентаційної розмітки не підтримуються в поточних рекомендаціях HTML і XHTML. Більшість презентаційних функцій з попередніх версій HTML більше не дозволені, оскільки вони призводять до погіршення

47

доступності, дорожчих витрат на обслуговування сайту та збільшення розмірів документів.[40]

Хороший семантичний HTML також покращує доступність веб-документів (див. також Інструкції щодо доступності веб-вмісту). Наприклад, коли програма зчитування з екрана або аудіобраузер можуть правильно визначити структуру документа, вони не витратять час користувача з вадами зору на зчитування повторюваної або невідповідної інформації, якщо вона була правильно позначена.

Каскадні таблиці стилів (CSS) — це мова таблиць стилів, яка використовується для опису подання документа, написаного мовою розмітки, як-от HTML.[1] CSS є наріжною технологією всесвітньої мережі, поряд з HTML і JavaScript.[32]

CSS розроблено, щоб дозволити розділити презентацію та вміст, включаючи макет, кольори та шрифти.[3] Це поділ може покращити доступність вмісту; забезпечити більшу гнучкість і контроль у специфікації характеристик презентації; дозволити кільком веб-сторінкам спільно використовувати форматування, вказавши відповідний CSS в окремому файлі .css, що зменшує складність і повторюваність структурного вмісту; і увімкніть кешування файлу .css, щоб покращити швидкість завантаження сторінки між сторінками, які спільно використовують файл, та його форматування.

Розділення форматування та вмісту також дає можливість представити одну й ту саму сторінку розмітки в різних стилях для різних методів візуалізації, наприклад, на екрані, у друкованому вигляді, голосом (через мовленнєвий браузер або програму зчитування з екрана) та на основі Брайля. тактильні пристрої. CSS також має правила альтернативного форматування, якщо доступ до вмісту здійснюється на мобільному пристрої.[34]

Каскадність назв походить від зазначеної схеми пріоритету, щоб визначити, яке правило стилю застосовується, якщо більше одного правила

48

відповідає певному елементу. Ця каскадна схема пріоритетів є передбачуваною.

Специфікації CSS підтримуються Консорціумом World Wide Web Consortium (W3C). Тип Інтернет-медіа (тип MIME) text/css зареєстровано для використання з CSS згідно RFC 2318 (березень 1998 р.). W3C надає безкоштовну службу перевірки CSS для документів CSS.[5]

На додаток до HTML, інші мови розмітки підтримують використання CSS, включаючи XHTML, звичайний XML, SVG і XUL.

Блок оголошення складається зі списку декларацій у дужках. Кожне оголошення складається з властивості, двокрапки (:) і значення. Якщо в блоці є кілька декларацій, для розділення кожної декларації потрібно вставити крапку з комою (;). Додаткова крапка з комою після останнього (або окремого) оголошення може використовуватися.[39]

Властивості вказані в стандарті CSS. Кожна властивість має набір можливих значень. Деякі властивості можуть впливати на будь-який тип елемента, а інші застосовуються лише до окремих груп елементів.[10][11]

Значеннями можуть бути ключові слова, наприклад «центр» або «успадкувати», або числові значення, наприклад 200 пікселів (200 пікселів), 50vw (50 відсотків ширини вікна перегляду) або 80% (80 відсотків ширини батьківського елемента). Значення кольору можна вказати за допомогою ключових слів (наприклад, "червоний"), шістнадцяткових значень (наприклад, #FF0000, також скорочено як #F00), значень RGB у шкалі від 0 до 255 (наприклад, rgb(255, 0, 0)), значень RGBA які вказують як колір, так і альфа-прозорість (наприклад, rgba(255, 0, 0, 0,8)), або значення HSL або HSLA (наприклад, hsl(000, 100%, 50%), hsla(000, 100%, 50%, 80 %)).[32]

Ненульові числові значення, що представляють лінійні міри, повинні включати одиницю довжини, яка є або алфавітним кодом, або аббревіатурою, як у 200px або 50vw; або знак відсотка, як у 80%. Деякі одиниці – см (сантиметр); в (дюйм); мм (міліметр); ПК (ріса); і pt (точка) – абсолютні, що

49

означає, що відтворений розмір не залежить від структури сторінки; інші – em (em); ex (ex) і px (піксель) [потрібне уточнення] – є відносними, що означає, що такі фактори, як розмір шрифту батьківського елемента, можуть вплинути на відтворене вимірювання. Ці вісім одиниць були властивістю CSS 1[33] і зберігалися в усіх наступних версіях. Запропонований модуль 3-го рівня значень та одиниць CSS, якщо його буде прийнято як рекомендацію W3C, надасть сім додаткових одиниць довжини: ch; Q; rem; vh; vmax; vmin; і vw.[34]

До CSS майже всі атрибути презентації HTML-документів містилися в розмітці HTML. Усі кольори шрифту, стилі фону, вирівнювання елементів, межі та розміри повинні бути чітко описані, часто неодноразово, у HTML. CSS дозволяє авторам переміщувати більшу частину цієї інформації в інший файл, таблицю стилів, що призводить до значно простішого HTML.

Наприклад, заголовки (елементи h1), підзаголовки (h2), підзаголовки (h3) тощо визначаються структурно за допомогою HTML. У друку та на екрані вибір шрифту, розміру, кольору та акцентування цих елементів є презентаційним.

До CSS автори документів, які хотіли призначити такі типографічні характеристики, скажімо, всім заголовкам h2, повинні були повторювати презентаційну розмітку HTML для кожного входження цього типу заголовка. Це зробило документи складнішими, більшими, більш схильними до помилок і ускладнювало обслуговування. CSS дозволяє відокремити презентацію від структури. CSS може визначати колір, шрифт, вирівнювання тексту, розмір, межі, інтервали, макет та багато інших типографічних характеристик, і може робити це незалежно для екранного та друкованого вигляду. CSS також визначає невізуальні стилі, такі як швидкість читання та наголос для читачів на слух. Тепер W3C відмовився від використання всієї презентаційної розмітки HTML.[35]

50

Наприклад, у попередньому CSS HTML елемент заголовка, визначений червоним текстом, буде записаний так:

```
<h1><font color="red">Розділ 1.</font></h1>
```

Використовуючи CSS, той самий елемент можна закодувати, використовуючи властивості стилю замість атрибутів HTML презентації:

```
<h1 style="color: red;">Розділ 1.</h1>
```

Переваги цього можуть бути не відразу зрозумілі, але потужність CSS стає більш очевидною, коли властивості стилю розміщені у внутрішньому елементі стилю або, ще краще, у зовнішньому файлі CSS. Наприклад, припустимо, що документ містить елемент стилю:

```
<СТИЛЬ>
```

```
h1 {  
    колір: червоний;  
}
```

```
</style>
```

Усі елементи h1 в документі автоматично стануть червоними, не вимагаючи жодного явного коду. Якщо пізніше автор захотів зробити елементи h1 синіми, це можна зробити, змінивши елемент стилю на:

```
<СТИЛЬ>
```

```
h1 {  
    колір: синій;  
}
```

```
</style>
```

замість того, щоб копітку переглядати документ і змінювати колір для кожного окремого елемента h1.

Стилі також можна помістити у зовнішній файл CSS, як описано нижче, і завантажити за допомогою синтаксису, схожого на:

51

```
<link href="path/to/file.css" rel="stylesheet" type="text/css">
```

Це ще більше відокремлює стиль від HTML-документа і дає можливість змінити стиль кількох документів, просто редагуючи спільний зовнішній файл CSS.

MySQL [35] — це система управління реляційними базами даних (RDBMS) з відкритим вихідним кодом.[36] Його назва є комбінацією «Му», імені дочки співзасновника Майкла Віденіуса [37], і «SQL», аббревіатури від Structured Query Language. Реляційна база даних організовує дані в одну або кілька таблиць даних, у яких типи даних можуть бути пов'язані один з одним; ці відносини допомагають структурувати дані. SQL — це мова, яку програмісти використовують для створення, зміни та вилучення даних з реляційної бази даних, а також для керування доступом користувачів до бази даних. На додаток до реляційних баз даних і SQL, СУБД, як-от MySQL, працює з операційною системою, щоб реалізувати реляційну базу даних у системі зберігання комп'ютера, керує користувачами, забезпечує доступ до мережі та полегшує перевірку цілісності бази даних і створення резервних копій.

MySQL є безкоштовним програмним забезпеченням з відкритим вихідним кодом згідно з умовами Загальної публічної ліцензії GNU, а також доступне на підставі різноманітних власних ліцензій. MySQL належав і спонсорувався шведською компанією MySQL AB, яку купила Sun Microsystems (тепер Oracle Corporation).[8] У 2010 році, коли Oracle придбала Sun, Widenius розширив проект MySQL з відкритим кодом для створення MariaDB.[9]

MySQL має автономні клієнти, які дозволяють користувачам безпосередньо взаємодіяти з базою даних MySQL за допомогою SQL, але частіше MySQL використовується з іншими програмами для реалізації програм, які потребують можливості реляційної бази даних. MySQL є компонентом стеку програмного забезпечення веб-додатків LAMP (та інших),

52

який є аббревіатурою для Linux, Apache, MySQL, Perl/PHP/Python. MySQL використовується багатьма веб-додатками, що керуються базами даних, включаючи Drupal, Joomla, phpBB і WordPress.

MySQL також використовується багатьма популярними веб-сайтами, включаючи Facebook, Flickr, MediaWiki, Twitter і YouTube.

MySQL написаний на C і C++. Його синтаксичний аналізатор SQL написаний на yacc, але він використовує саморобний лексичний аналізатор. [16] MySQL працює на багатьох системних платформах, включаючи AIX, BSDi, FreeBSD, HP-UX, ArcaOS, eComStation, IBM і, IRIX, Linux, macOS, Microsoft Windows, NetBSD, Novell NetWare, OpenBSD, OpenSolaris, OS/2 Warp, QNX, Oracle Solaris, Symbian, SunOS, SCO OpenServer, SCO UnixWare, Sanos і Tru64. Також існує порт MySQL до OpenVMS.[26]

Саме серверне програмне забезпечення MySQL і клієнтські бібліотеки використовують дистрибутив з подвійним ліцензуванням. Вони пропонуються під GPL версії 2 або за власною ліцензією.[18]

Підтримку можна отримати в офіційному посібнику.[19] Крім того, безкоштовна підтримка доступна на різних каналах IRC і на форумах. Oracle пропонує платну підтримку через свої продукти MySQL Enterprise. Вони відрізняються за обсягом послуг і ціною. Крім того, існує ряд сторонніх організацій, які надають підтримку та послуги.

MySQL отримав позитивні відгуки, і рецензенти помітили, що він «надзвичайно добре працює в середньому», і що «інтерфейси розробника є, а документація (не кажучи вже про зворотний зв'язок у реальному світі через веб-сайти тощо) дуже, дуже добре».[28] Він також був протестований як «швидкий, стабільний і справжній багатокористувацький, багатопоточний сервер баз даних SQL».[28]

MySQL можна зібрати та встановити вручну з вихідного коду, але частіше він встановлюється з бінарного пакета, якщо не потрібні спеціальні налаштування. У більшості дистрибутивів Linux система керування пакетами

53

може завантажити та встановити MySQL з мінімальними зусиллями, хоча для налаштування параметрів безпеки та оптимізації часто потрібна додаткова конфігурація.

Незважаючи на те, що MySQL починався як альтернатива більш потужним власним базам даних, він поступово розвивався для підтримки потреб більшого масштабу. Він як і раніше найчастіше використовується в малих і середніх розгортаннях з одним сервером, або як компонент у веб-додатку на основі LAMP, або як окремий сервер бази даних. Значна частина привабливості MySQL походить від її відносної простоти та зручності використання, що забезпечується екосистемою інструментів з відкритим кодом, наприклад phpMyAdmin. У середньому діапазоні MySQL можна масштабувати, розгортаючи його на більш потужному обладнанні, такому як багатопроцесорний сервер з гігабайтами пам'яті.

Однак існують обмеження щодо того, наскільки продуктивність може розширюватися на одному сервері («збільшення масштабу»), тому для більших масштабів для підвищення продуктивності та надійності потрібні багатосерверні розгортання MySQL («масштабування»). Типова конфігурація високого класу може включати потужну головну базу даних, яка обробляє операції запису даних і реплікується на кілька ведених, які обробляють усі операції читання.[30] Головний сервер постійно передає події binlog на підключені підпорядковані пристрої, тому в разі збою підпорядкованого можна підвищити до нового головного, що мінімізує час простою.

Подальшого покращення продуктивності можна досягти шляхом кешування результатів запитів до бази даних у пам'яті за допомогою memcached або розбиття бази даних на менші фрагменти, які називаються сегментами, які можуть бути розподілені між кількома розподіленими серверними кластерами.[39]

## РОЗДІЛ 3

### ПРОЕКТУВАННЯ І РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ

#### 3.1 Аналіз варіантів використання

Діаграма використання найпростіша - це представлення взаємодії користувача із системою, яка показує взаємозв'язок між користувачем та різними випадками використання, в яких користувач бере участь. Діаграма випадків використання може ідентифікувати різні типи користувачів системи та різні випадки використання, і часто вона супроводжується також іншими типами діаграм. Варіанти використання представлені кругами або еліпсами.

Незважаючи на те, що сам випадок використання може детально вивчити кожну можливість, діаграма прикладів використання може допомогти забезпечити огляд системи на більш високому рівні. Раніше вже було сказано, що "схеми використання - це принципи вашої системи".

Через їх спрощений характер, схеми використання можуть бути хорошим інструментом комунікації для зацікавлених сторін. Креслення намагаються імітувати реальний світ і дають зацікавленій стороні уявлення про те, як буде розроблена система. Сіау та Лі провели дослідження, щоб визначити, чи взагалі існувала дійсна ситуація для схем використання або вони були непотрібними. Було виявлено, що діаграми випадків використання передають намір системи більш спрощеним чином зацікавленим сторонам і що вони "інтерпретуються більш повно, ніж діаграми класів".

Метою діаграми використання є відображення динамічного аспекту системи. Додаткові схеми та документація можуть бути використані для забезпечення повного функціонального та технічного уявлення про систему. Вони забезпечують спрощене та графічне представлення того, що система насправді повинна робити.

Елементи:

55

- рамки системи (англ. system border) - прямокутник із назвою у верхніх частинах та еліпсами (прецедентами) всередині. Часто може бути опущено без корисної інформації про полезну інформацію,
- актор (англ. actor) - стилізований людський персонаж, обзначаючий набір ролей користувача (розуміється в широкому змісті: людина, зовнішня сутність, клас, інша система), взаємодіючого з деякою сутністю (системною, підсистемою, класом). Актори не можуть бути пов'язані між собою з іншим (за вимкнення відносин щодо обробки / дослідження),
- прецедент - еліпс із надписом, що означає виконувану систематичну дію (може включати можливі варіанти), що призводить до спостережуваних акторами результатів. Надпис може бути ім'ям або описом (з точки зору актора) того, "що" робить система (а не "як"). Ім прецедента зв'язано з неперервним (атомарним) сценарієм - конкретною послідовністю дій, ілюструючою поведінку. Під час сценарію актори обмінюються із систематичними повідомленнями. Сценарій може бути приведений на діаграмі прецедентів у відео UML-коментарі. З одним прецедентом може бути пов'язано кілька різних сценаріїв

На рисунку 3.1 зображено діаграму варіантів використання, яка описує можливі дії користувача в системі.

56



Рис. 3.1. Діаграма варіантів використання

### 3.2 Проектування внутрішньої будови

У програмній інженерії діаграма класів в Уніфікованій мові моделювання (UML) - це тип статичної структурної діаграми, що описує структуру системи, показуючи класи системи, їх атрибути, операції (або методи) та взаємозв'язки між об'єктами.

Діаграма класів є основним будівельним елементом об'єктно-орієнтованого моделювання. Він використовується для загального концептуального моделювання структури програми та для детального моделювання переведення моделей у програмовий код. Діаграми класів також можуть бути використані для моделювання даних. Класи на діаграмі класів представляють як основні елементи, взаємодії в програмі, так і класи, що програмуються.

На схемі класи представлені вікнами, які містять три відділення:

- У верхньому відділенні міститься назва класу. Надруковано жирним шрифтом і відцентровано, а перша літера написана великими літерами.

57

- Середній відсік містить атрибути класу. Вони вирівняні за лівим краєм, а перша буква мала.
- У нижньому відділенні містяться операції, які може виконувати клас. Вони також вирівняні за лівим краєм, а перша буква - мала.

При проектуванні системи ряд класів ідентифікується та згруповується у схему класів, яка допомагає визначити статичні відносини між ними. При детальному моделюванні класи концептуального проекту часто поділяються на ряд підкласів.

Залежність - це семантичний зв'язок між залежними та незалежними елементами моделі. Він існує між двома елементами, якщо зміни у визначенні одного елемента (сервера або цілі) можуть спричинити зміни для іншого (клієнта або джерела). Ця асоціація є односпрямованою. Залежність відображається у вигляді штрихової лінії з відкритою стрілкою, яка вказує від клієнта до постачальника.

Для подальшого опису поведінки систем ці діаграми класів можуть бути доповнені діаграмою стану або машиною стану UML.

Асоціація представляє родину посилань. Двійкова асоціація (з двома кінцями) зазвичай представляється у вигляді рядка. Асоціація може пов'язувати будь-яку кількість класів. Асоціація з трьома ланками називається потрійною асоціацією. Асоціацію можна назвати, а кінці асоціації можна прикрасити іменами ролей, показниками власності, кратністю, видимістю та іншими властивостями.

Існує чотири різні типи асоціацій: двонаправлена, односпрямована, агрегаційна (включає агрегацію композиції) та рефлексивна. Двонаправлені та односпрямовані асоціації є найбільш поширеними.

Наприклад, клас польоту асоціюється з класом літака двонаправлено. Асоціація представляє статичне відношення, яке ділиться між об'єктами двох класів.

58

Агрегація є варіантом взаємозв'язку "має"; агрегація є більш конкретною, ніж асоціація. Це асоціація, яка представляє частково цілі або часткові стосунки. Як показано на зображенні, професор "має" клас для викладання. Як тип асоціації, агрегація може бути названа та мати ті самі прикраси, що і асоціація. Однак агрегація не може включати більше двох класів; це має бути бінарна асоціація. Крім того, навряд чи існує різниця між агрегаціями та асоціаціями під час реалізації, і діаграма може взагалі пропустити відносини агрегування. [7]

Агрегація може відбуватися, коли клас є колекцією або контейнером інших класів, але вміщені класи не мають сильної залежності життєвого циклу від контейнера. Вміст контейнера все ще існує, коли контейнер знищений.

В UML він графічно представлений у вигляді порожньої форми ромба на вміщуючому класі одним рядком, що зв'язує його із вміщеним класом. Сукупність - це семантично розширений об'єкт, який у багатьох операціях трактується як одиниця, хоча фізично він складається з декількох менших об'єктів.

Приклад: Бібліотека та студенти. Тут студент може існувати без бібліотеки, зв'язок між студентом і бібліотекою є агрегацією.

Це вказує на те, що один із двох пов'язаних класів (підклас) вважається спеціалізованою формою іншого (супер тип), а суперклас - узагальненням підкласу. На практиці це означає, що будь-який екземпляр підтипу є також екземпляром суперкласу. Зразкове дерево узагальнень цієї форми зустрічається в біологічній класифікації: людина - це підклас маймуни, який є підкласом ссавців тощо. Зв'язок найлегше зрозуміти за допомогою фрази „А - це В” (людина - це ссавець, ссавець - тварина).

Графічне представлення UML узагальнення - це форма порожнього трикутника на кінці суперкласу рядка (або дерева рядків), що зв'язує його з одним або кількома підтипами.

59

Відносини узагальнення також відомі як спадщина або відносини "є".

Суперклас (базовий клас) у відносинах узагальнення також відомий як "батьківський", суперклас, базовий клас або базовий тип.

Підтип у відносинах спеціалізації також відомий як "дочірній", підклас, похідний клас, похідний тип, клас успадкування або тип успадкування.

Зверніть увагу, що ці стосунки нічим не схожі на біологічні стосунки батьків та дітей: використання цих термінів надзвичайно поширене, але може ввести в оману.

**A - це тип B**

Наприклад, "дуб - це тип дерева", "автомобіль - це тип транспортного засобу"

Узагальнення може бути показано лише на діаграмах класів та на діаграмах використання.

При моделюванні UML взаємозв'язок реалізації - це взаємозв'язок між двома елементами моделі, в яких один елемент моделі (клієнт) реалізує (реалізує або виконує) поведінку, яку вказує інший елемент моделі (постачальник).

Графічне представлення UML реалізації - це порожниста форма трикутника на кінці інтерфейсу штрихової лінії (або дерева рядків), яка з'єднує її з одним або кількома реалізаторами. Проста головка стрілки використовується на кінці інтерфейсу штрихової лінії, що з'єднує її з користувачами. У діаграмах компонентів використовується графічна умова «м'яч і сокет» (реалізатори виставляють кульку або льодяник, тоді як користувачі показують сокет). Реалізації можна показати лише на діаграмах класів або компонентів. Реалізація - це взаємозв'язок між класами, інтерфейсами, компонентами та пакетами, що з'єднує елемент клієнта з елементом постачальника. Зв'язок реалізації між класами / компонентами та інтерфейсами показує, що клас / компонент реалізує операції, пропонувані інтерфейсом.

60

Залежність - це слабша форма зв'язку, яка вказує на те, що один клас залежить від іншого, оскільки він використовує його в певний момент часу. Один клас залежить від іншого, якщо незалежний клас є змінною параметра або локальною змінною методу залежного класу. Це відрізняється від асоціації, де атрибут залежного класу є екземпляром незалежного класу. Іноді відносини між двома класами дуже слабкі. Вони взагалі не реалізовані зі змінними-членами, але можуть бути реалізовані як аргументи функції-члена.

Представлення UML асоціації - це лінія, що з'єднує два пов'язані класи. На кожному кінці рядка є додаткові позначення. Наприклад, ми можемо вказати, використовуючи наконечник стрілки, що загострений кінець видно з хвоста стрілки. Ми можемо вказати власність шляхом розміщення кульки, ролі, яку відіграють елементи цього кінця, вказавши ім'я ролі та множинність екземплярів цієї сутності (діапазон кількості об'єктів, які беруть участь в асоціації з точки зору іншого кінця).

Класи сутності моделюють довгоживучу інформацію, якою обробляє система, а іноді і поведінку, пов'язану з цією інформацією. Їх не слід ідентифікувати як таблиці баз даних чи інших сховищ даних.

Вони намальовані як кола з короткою лінією, прикріпленою до нижньої частини кола. Як варіант, їх можна намалювати як звичайні класи із позначенням стереотипу «сутність» над назвою класу.

61



Рис. 3.2. Діаграма класів

62

### 3.3 Розробка графічного інтерфейсу системи

Графічний інтерфейс системи складається з таких основних сторінок:

- сторінка реєстрації (рис. 3.3) — початкова сторінка веб-сервісу, яка дозволяє зареєструвати новий акаунт або перейти на сторінку входу в вже існуючий за допомогою кнопки «Вже маєте акаунт?».

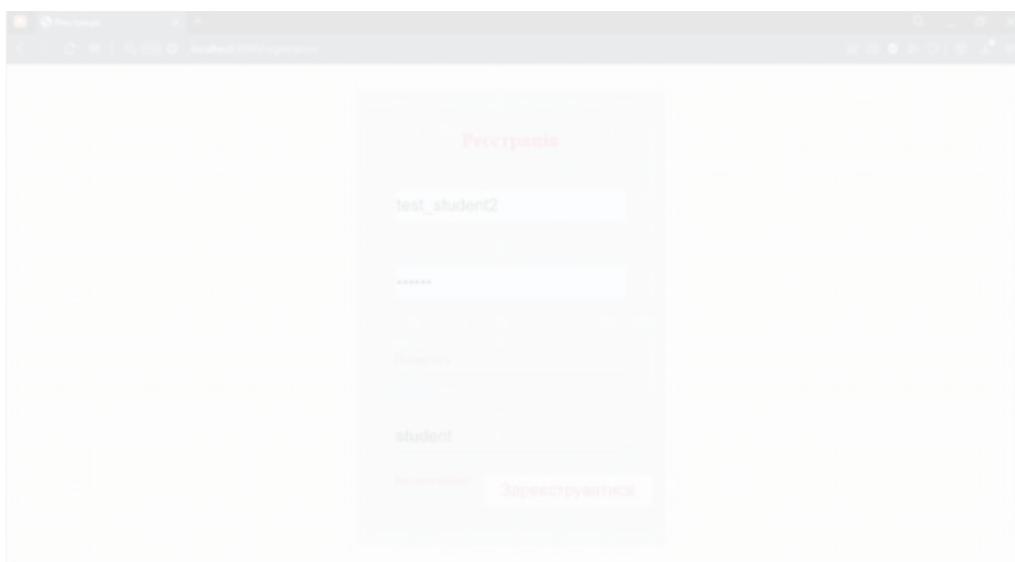


Рисунок 3.3. Сторінка реєстрації

Реєстрацію нового користувача було розроблено наступним чином:

```
public String registration(Model model)
{
    model.addAttribute("title", "Реєстрація");
    if(!authorized)
    {
        return "registration";
    }
    else
    {
        return "redirect:/";
    }
}
```

63

```
}  
}
```

- сторінка авторизації (рис. 3.4) — дозволяє здійснити вхід в вже існуючий акаунт за допомогою даних, введених при реєстрації раніше. Кнопка «Не маєте акаунту?» повертає на сторінку реєстрації нового акаунту системи.



Рисунок 3.4. Сторінка авторизації

Авторизацію користувача забезпечує наступний фрагмент коду Java:

```
public String authorize(Model model, @RequestParam String user_login,  
{  
    Iterable<Users> users_it = usersRepository.findAll();  
    List<Users> users = new ArrayList<>();  
    users_it.forEach(users::add);  
    for (int i = 0; i < users.size(); i++)  
    {  
        if(users.get(i).getUser_login().equals(user_login) &&  
users.get(i).getUser_password().equals(user_password))  
        {  
            authorized = true;
```

64

```

        current_user = users.get(i);
        break;
    }
}
return "redirect:";
}

```

- головна сторінка (рис. 3.5) — початкова сторінка Веб-сервісу агентства знайомств, перехід на неї можливий у разі успішного входу в акаунт так містить меню навігації вгорі сторінки.

Графічний інтерфейс навігаційного меню було розроблено наступними засобами мови розмітки гіпертексту HTML:

```

<div th:fragment="header_user">
    <div class="d-flex flex-column flex-md-row align-items-center pb-3 mb-4
border-bottom">
        <a href="/" class="d-flex align-items-center text-dark text-decoration-none">
            <span class="fs-4">Агенство знайомств</span>
        </a>
        <nav class="d-inline-flex mt-2 mt-md-0 ms-md-auto">
            <a class="me-3 py-2 text-dark text-decoration-none"
href="/">Головна</a>
            <a class="me-3 py-2 text-dark text-decoration-none"
href="/mans">Чоловіки</a>
            <a class="me-3 py-2 text-dark text-decoration-none"
href="/womens">Жінки</a>
            <a class="me-3 py-2 text-dark text-decoration-none"
href="/pairs">Пари</a>
            <a class="py-2 text-dark text-decoration-none" href="/exit">Вихід</a>
        </nav>
    </div>

```

65

</div>



Рисунок 3.5. Головна сторінка

Кожна кнопка навігаційного меню веде до однойменної сторінки, надаючи повний доступ користувачу до всього спектру функціоналу.

- сторінка чоловічих анкет (рис. 3.6) — дозволяє переглядати всі чоловічі анкети наявні в базі агентства на даний момент.



Рисунок 3.6. Сторінка чоловічих анкет

66

- сторінка жіночих анкет (рис. 3.7) — дозволяє переглядати всі жіночі анкети наявні в базі агентства на даний момент.

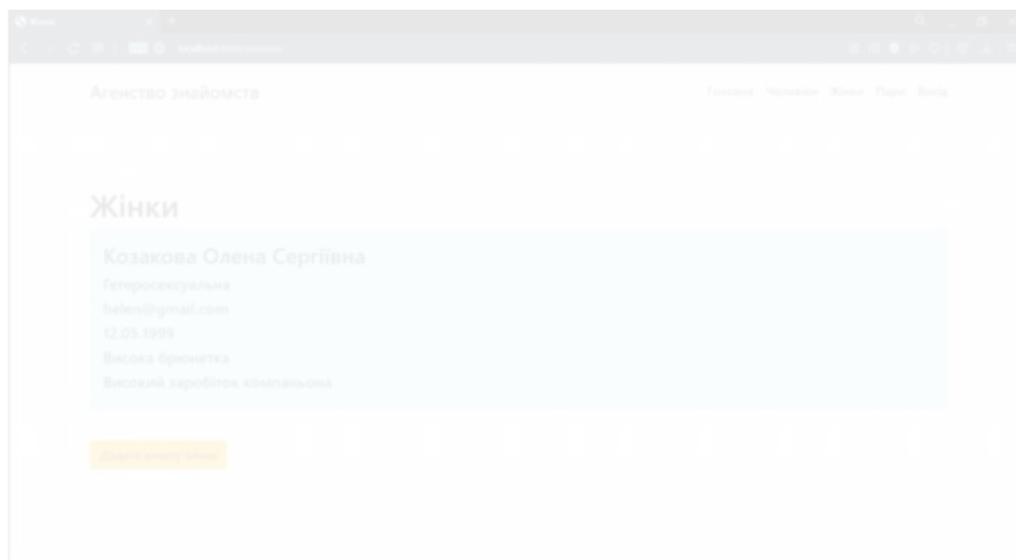


Рисунок 3.7. Сторінка жіночих анкет

- сторінка пар (рис. 3.8) — дозволяє переглядати створені пари. Містить кнопку «Додати пару», що відкриває сторінку додавання нової пари (рис. 3.11).

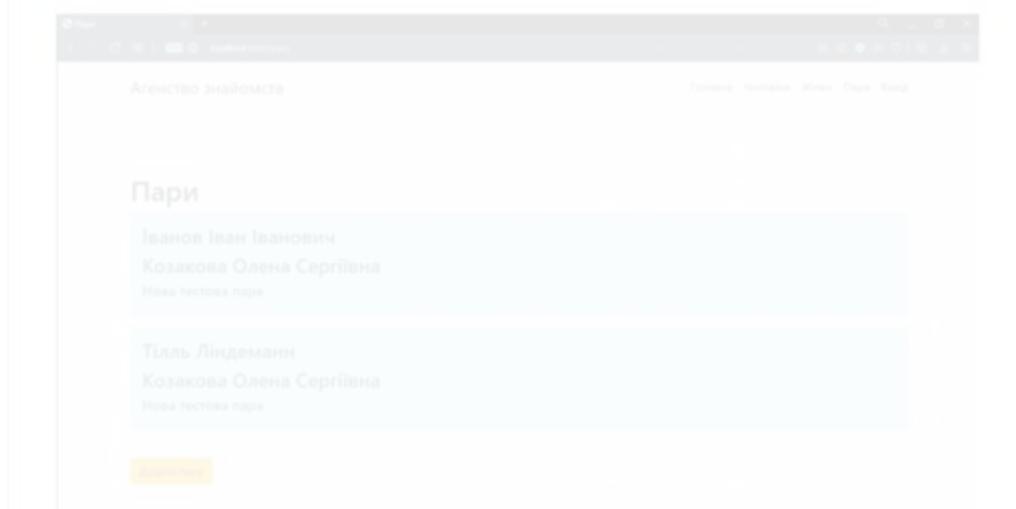


Рисунок 3.8. Сторінка пар

67

- сторінка додавання жіночої анкети (рис. 3.9) — дозволяє додати нову жіночу анкету з введенням необхідної інформації про жінку, яка буде використовуватись для створення пари.

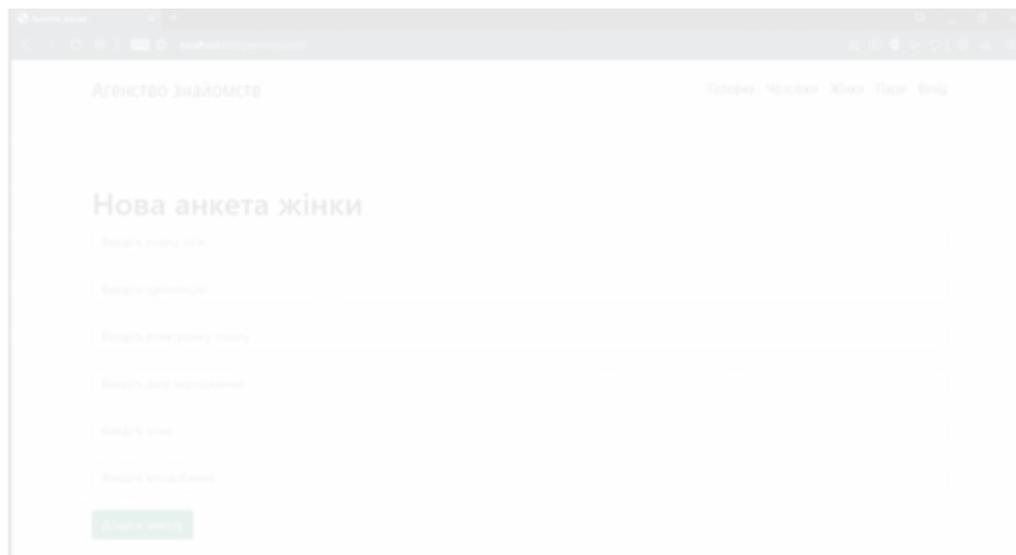


Рисунок 3.9. Сторінка додавання жіночої анкети

Створення нової анкети жінки з всіма необхідними даними (повне ім'я, орієнтація, електронна пошта, дата народження, опис та вподобання) забезпечує наступний рядок коду:

```
Womens women = new Womens(full_name, orientation, email, dob, description, preferences);
```

- сторінка додавання чоловічої анкети (рис. 3.10) — аналогічно Сторінці додавання жіночої анкети, дозволяє додати нову чоловічу анкету.

Нова анкета чоловіка додається аналогічно до жіночої, за допомогою наступного фрагмента коду:

```
Mans man = new Mans(full_name, orientation, email, dob, description, preferences);
```

68

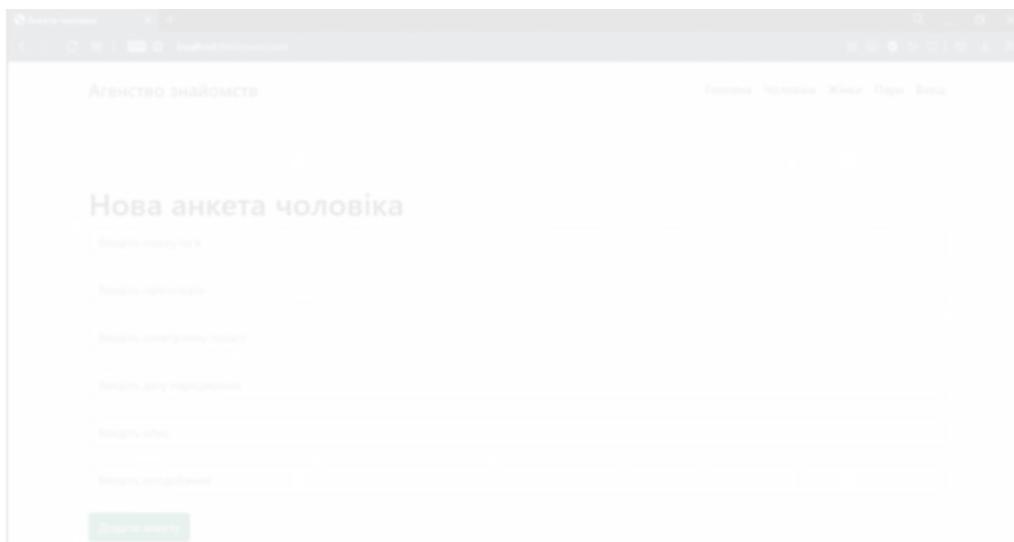


Рисунок 3.10. Сторінка додавання чоловічої анкети

- сторінка додавання пари (рис. 3.11) — дозволяє додати нову пару.

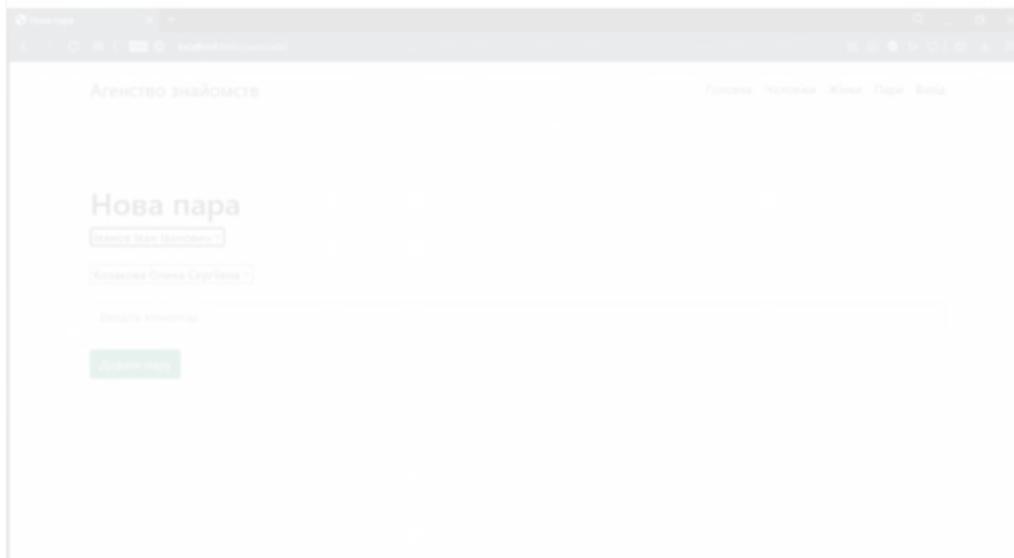


Рисунок 3.11. Сторінка додавання пари

Нова пара додається за допомогою наступного розробленого коду:

```
public String pairsAdd(Model model) {  
    Iterable<Womens> womens_it = womenRepository.findAll();
```

69

```
List<Womens> womens = new ArrayList<>();  
womens_it.forEach(womens::add);  
  
Iterable<Mans> mans_it = manRepository.findAll();  
List<Mans> mans = new ArrayList<>();  
mans_it.forEach(mans::add);  
model.addAttribute("mans", mans);  
model.addAttribute("womens", womens);  
model.addAttribute("title", "Нова пара");  
return "pairs_add";  
}
```

70

## ВИСНОВКИ

Під час виконання кваліфікаційної роботи було проведено огляд діяльності агентства знайомств та встановлені їх основні функції в сучасних умовах. Для досягнення поставленої мети було проведено огляд сучасної агентств знайомств на сьогоднішній день, а також в минулі часи.

Було здійснено огляд інформаційної системи підприємства з надання послуг, дослідження методів розробки веб-сервісів, було розроблено Веб-сервіс для інформаційної підтримки діяльності агентства знайомств за допомогою мови Java, використовуючи інтегроване середовище розробки IntelliJ IDEA. В ході роботи також було оглянуто основні функції мови програмування та інструментарій середовища розробки з подальшим їх використанням.

Графічний інтерфейс веб-сервісу було розроблено за допомогою мови розмітки гіпертексту або HTML та мови таблиць стилів CSS.

Серед функціоналу веб-сервісу було розроблено:

- головну сторінку;
- можливість реєстрації користувача з введенням власних даних;
- перегляд жіночих та чоловічих анкет;
- додавання власної анкети з вказанням своєї інформації та критеріїв підбору пари;
- перегляд створених пар;
- створення пари за всіма вказаними критеріями відбору.

Завдяки чіткому виконанню завдань, поставлених на початку роботи, в результаті отримано повноцінну систему контролю процесів агентства знайомств, яка готова до використання в реальних умовах. Одержані результати дають підстави вважати, що поставлену мету було виконано.

Веб-сервіс може вдосконалюватись в подальшому, видозмінюватись для більш зручного та ефективного використання методом розширення списку функцій для задоволення потреб більшої кількості користувачів.

# Диплом.СельвесюкІ.І.

Завантажено: 06/19/2022 | Перевірено: 06/19/2022

● Схожість   ● Цитата   ● Використані джерела   ● Заміна символів

71

### ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Stephen Jones. "New Matrimonial Plan", The Spirit of the Public Journals for 1799: Being an Impartial Selection of the Most Exquisite Essays and Jeux d'Esprits, Particularly Prose, That Appear in the Newspapers and Other Publications: Видання. Palala Press, 2008. 329 с
2. William Hone. "A New Matrimonial Plan", The Every-day Book and Table Book: Видання. London, 1838, 22-23 сс.
3. FBI Cautions Public to be Wary of Online Romance Scams. URL: <https://www.houstonpublicmedia.org/articles/news/criminal-justice/2022/02/11/418897> (дата звернення 01.06.2022).
4. Marriage agency - Wiktionary. URL: [en.wiktionary.org](https://en.wiktionary.org) (дата звернення 01.06.2022)..
5. The evolution of: matchmaking - Reader's Digest. URL: [www.readersdigest.co.uk](http://www.readersdigest.co.uk) (дата звернення 01.06.2022).
6. Social and Demographic Trends Team . URL: <https://www.pewresearch.org/about/research-teams/social-and-demographic-trends-team/> (дата звернення 01.06.2022).
7. New Partners, More Kids: Multiple-Partner Fertility in the United States URL: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4182921/12-14> (дата звернення 01.06.2022).
8. America Is a Singles Nation, Or Is It? URL: <https://www.psychologytoday.com/us/blog/living-single/201506/america-is-singles-nation-or-is-it> (дата звернення 01.06.2022).
9. NowUKnow: Why Millennials Refuse to Get Married | Bentley University. URL: [www.bentley.edu](http://www.bentley.edu) (дата звернення 01.06.2022).
10. Japan's Unmarried Masses Face Mounting Obstacles to Matrimony. URL: [www.nippon.com](http://www.nippon.com) (дата звернення 01.06.2022).

73

11. Jackson, Molly. "How Americans' views of marriage have changed over time". URL: <https://www.businessinsider.com/how-americans-views-of-marriage-have-changed-over-time-2016-2> (дата звернення 01.06.2022).
12. Lichter, Daniel T.; Price, Joseph P.; Swigert, Jeffrey M. Mismatches in the Marriage Market. Journal of Marriage and Family: Видання. Lichter, Daniel T., 2020, 509-514.
13. Are Dating Apps Damaging Our Mental Health? URL: <https://www.psychologytoday.com> (дата звернення 01.06.2022).
14. Dating Agency. URL: <https://bearl.one/> (дата звернення 01.06.2022).
15. Message from CEO | IBJ Inc. URL: [www.ibj-gl.com](http://www.ibj-gl.com) (дата звернення 01.06.2022).
16. National Institute of Population. Marriage Process and Fertility of Married Couples Attitudes toward Marriage and Family among Japanese Singles: Статистичні дані. National Institute of Population, 2017, 27 с.
17. Ryan W. Buell, Michael I. Norton. The Labor Illusion: How Operational Transparency Increases Perceived Value: Наукова публікація. Harvard Business School, Harvard University, Boston, Massachusetts, 2017, 1564–1579 сс
18. The "Ehime Method": Using Big Data to Support Matchmaking. URL: [nippon.com](http://nippon.com) (дата звернення 01.06.2022).
19. Marriage Abroad. URL: [travel.state.gov](http://travel.state.gov). (дата звернення 01.06.2022).
20. Getting married abroad. URL: [GOV.UK](http://GOV.UK) (дата звернення 01.06.2022).
21. Marriage Agency | Marriage Matching Marriage Agency. URL: <https://marriagematching.love/marriage-agency/> (дата звернення 01.06.2022).
22. "International Marriage Agency - ELITE. URL: [www.magzter.com](http://www.magzter.com). (дата звернення 01.06.2022).

74

23. 8 facts about love and marriage in America. URL: <https://www.pewresearch.org/fact-tank/2019/02/13/8-facts-about-love-and-marriage/> (дата звернення 01.06.2022).
24. Murphy, Hannah. Virtual dating on the rise as users seek lockdown love, says Match chief. URL: <https://www.ft.com/content/f970c353-65e3-4515-a179-e94b89ce3775> (дата звернення 01.06.2022).
25. Marriage agency URL: [https://en.everybodywiki.com/Marriage\\_agency](https://en.everybodywiki.com/Marriage_agency) (дата звернення 01.06.2022).
26. Брайан Глік. Писати один раз, бігти куди завгодно? Комп'ютерний тижневик. URL: <https://www.computerweekly.com/feature/Write-once-run-anywhere> (дата звернення 01.06.2022).
27. The Java Language Environment URL: <https://www.oracle.com/java/technologies/introduction-to-java.html> (дата звернення 01.06.2022)
28. Чан Розалі. 10 найпопулярніших мов програмування за версією Facebook для програмістів URL: <https://www.businessinsider.de/the-10-most-popular-programming-languages-according-to-github-2018-10?op=1>. (дата звернення 01.06.2022)
29. Писати один раз, бігти куди завгодно? Комп'ютерний тижневик. URL: <https://www.computerweekly.com/feature/Write-once-run-anywhere>. (дата звернення 01.06.2022)
30. Конспект лекцій з навчальної дисципліни «Розподілені сервісні системи» для студентів всіх форм навчання спеціальності «Телекомунікації та радіотехніка» відділу Інфокомунікацій та інженерії. / Уклад.: О.Д. Архелюк – Чернівці.: ЧНУ імені Юрія Федьковича 2021. – 94 с.
31. Єлович, Деян. Чому Java завжди буде повільніше, ніж C ++. URL: [https://uk.wikisu.ru/wiki/java\\_\(programming\\_language\)](https://uk.wikisu.ru/wiki/java_(programming_language)). (дата звернення 01.06.2022)

75

32. Java 2 Platform Enterprise Edition. URL:  
<https://www.oracle.com/java/technologies/>. (дата звернення 01.06.2022)
33. Вибір першої мови програмування — огляд Java. URL:  
<https://blog.ithillel.ua/articles/vybir-pershoi-movy-prohramuvannia-ohliad-java>. (дата звернення 01.06.2022).
34. Копитко М.Ф., Іванків К.С. Основи програмування мовою Java: Тексти лекцій. – Львів: Видавничий центр ЛНУ ім. Івана Франка, 2002.– 83 с.
35. IntelliJ IDEA :: рефакторинг Java плюс складний рефакторинг коду для JSP, XML, CSS, HTML, JavaScript. URL:  
<https://www.jetbrains.com/idea/features/refactoring.html>. (дата звернення 01.06.2022).
36. К.Арнольд, Дж.Гослінг. Мова програмування Java/Пер.с англ.-Санкт-Петербург, 1997.-304 с.
37. Керування пам'яттю у Java Script. URL:  
<https://codeguida.com/post/1045>. (дата звернення 01.06.2022).
38. К.Арнольд, Дж.Гослінг. Мова програмування Java/Пер.с англ.-СПб: Пітер, 1997.-304 с.
- 39.What is Memcached? URL: <http://memcached.org/> дата звернення 01.06.2022).
40. Методичні вказівки до лабораторних робіт "Створення веб-сайту за допомогою HTML" з курсу "Основи Internet-технологій" : для студ. спец. 124 "Системний аналіз", 186 "Видавництво і поліграфія" / уклад. А. Н. Марченко, В. О. Колбасін, В. П. Прокопенков ; Харківський політехнічний ін-т, нац. техн. ун-т. – Харків : НТУ "ХПІ", 2017. – 67 с.