

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Інститут післядипломної освіти

Кафедра інформаційних технологій

Кваліфікаційна робота бакалавра

на тему: Розробка освітнього веб ресурсу для вивчення мов програмування

Виконав студент групи КН-5
спеціальності 122 «Комп'ютерні науки»
Ахмедов Хазрет Ісмет огли

Керівник к.геогр.н., доцент
Кузніченко Світлана Дмитрівна

Консультант _____

Рецензент к.техн.н., доцент
Гнатовська Ганна Арнольдівна

Одеса 2023

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ	6
ВСТУП	7
1 ОГЛЯД ЗАСОБІВ ТА ТЕХНОЛОГІЙ РОЗРОБКИ ОСВІТНЬОГО ВЕБ-РЕСУРСУ	9
1.1 Порівняння систем аналогів.....	9
1.1.1 Ресурс Udacity для вивчення мов програмування	9
1.1.2 Ресурс Coursera для онлайн навчання.....	10
1.1.3 Ресурс Khan Academy для онлайн навчання	11
1.1.4 Ресурс EdX для онлайн навчання	12
1.1.5 Порівняльний аналіз функціоналу систем аналогів.....	13
1.2 Вибір технологій для розробки системи.....	14
1.2.1 Вибір мови програмування	14
1.2.2 Вибір та обґрунтування бази даних	15
1.2.3 Опис стеку технологій	17
2 ПРОЄКТУВАННЯ ОСВІТНЬОГО РЕСУРСУ	20
2.1 Мета та задачі інформаційної системи	20
2.2 Керування вимогами.....	22
2.2.1 Типи користувачів.....	22
2.2.2 Функціональні вимоги.....	22
2.2.3 Нефункціональні вимоги.....	26
2.3 Представлення користувальницького інтерфейсу.....	27
2.6 Логічне представлення системи	30
2.7 Представлення розгортання ІС	33
2.8 Представлення шарів ІС	34
2.9 Представлення процесів ІС	38
2.10 Представлення даних ІС	43
3 РЕАЛІЗАЦІЯ ОСВІТНЬОГО РЕСУРСУ	50
3.1 Уявлення про структуру проекту ІС	50

	5
3.2 Структура класів ІС	52
3.3 Інфраструктурне представлення ІС.....	53
ВИСНОВКИ.....	54
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	55
Додаток А Макети графічного інтерфейсу.....	58
Додаток Б Керівництво користувача.....	66

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

БД – база даних

ОС – операційна система

ПЗ – програмне забезпечення

ADT – Android Development Tools – Інструменти розробки в ОС Android

API – Application Programming Interface – програмний інтерфейс
програми

AR – Augmented Reality – доповнена реальність

IDE – Integrated Development Environment – інтегроване середовище
розробки

SDK – Software Development Kit – набір засобів розробки

ВСТУП

В наш час людям доводиться обробляти велику кількість інформації, тому вони вимушені шукать способи економії часу. Якщо порахувати, то в черзі та в транспорті ми витрачаємо багато часу, який могли би потратити на щось більш важливе. Сучасному студенту неможливо впоратися з колосальним потоком інформації, який щодня обрушується на нього. Навіть високорозвинена пам'ять не в змозі зберегти гігантський інформаційний масив. Тому з'являються нові завдання в навчанні, пов'язані з чітким відбором навчального матеріалу, структуруванням загальноосвітнього курсу програмування, складання удосконалених методик вивчення, що дозволяють за малу кількість часу отримувати максимум інформації.

Освітні веб-ресурси надають можливість студентам ознайомитися з основами програмування та отримати можливість самопідготовки до контролюючих засобів з відповідних дисциплін. Розробка подібного ресурсу може бути одним з елементів вдосконалення освітнього процесу підготовки спеціалістів з ІТ в коледжах та університетах. Реалізація зручного інтерфейсу веб-застосування зробить процес навчання більш простим та зрозумілим для початківців програмістів.

Метою кваліфікаційної роботи є розробка освітнього веб ресурсу для вивчення мов програмування.

Для досягнення поставленої мети в роботі необхідно вирішити наступні завдання:

- аналіз подібних існуючих освітніх веб-ресурсів для вивчення мов програмування;
- обґрунтування вибору засобів та технологій розробки програмного забезпечення;
- проектування інформаційної системи;
- реалізація програмної інформаційної системи;

– тестування програмного забезпечення та аналіз отриманих результатів.

Структура кваліфікаційної роботи складається вступу, трьох розділів, висновків, переліку посилань на 20 найменувань, додатків. Повний обсяг роботи становить 69 сторінок, містить 25 рисунків і 1 таблицю.

1 ОГЛЯД ЗАСОБІВ ТА ТЕХНОЛОГІЙ РОЗРОБКИ ОСВІТНЬОГО ВЕБ-РЕСУРСУ

1.1 Порівняння систем аналогів

1.1.1 Ресурс Udacity для вивчення мов програмування

Udacity – відомий ресурс для навчання програмуванню [1] – випустив безкоштовний додаток, завдяки якому користувач може істотно підвищити рівень своєї освіти, проходячи безкоштовні курси з програмування від провідних експертів ІТ-галузі (рис.1.1).

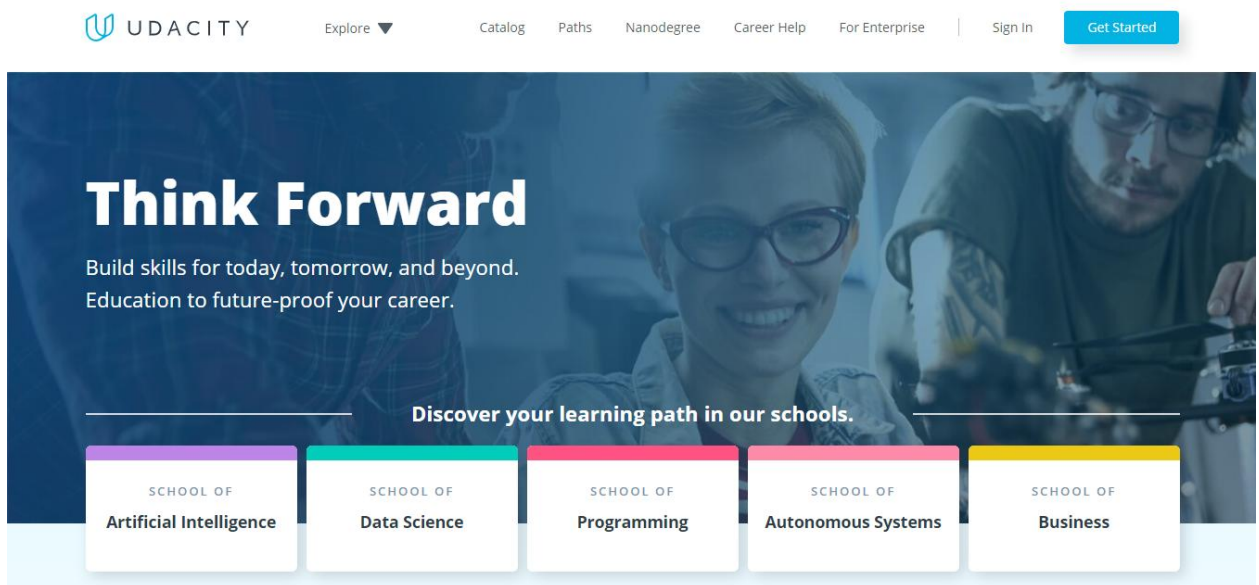


Рисунок 1.1 – Головна сторінка ресурсу Udacity

Відео лекції англійською мовою з субтитрами в поєднанні з вбудованими тестами і подальшими домашніми роботами, засновані на моделі «вчитися на практиці». Кожна лекція включає в себе вбудований тест, щоб допомогти студентам зрозуміти пропоновані концепції та ідеї. З іншого боку, на форумах досі багато незадоволених: курси, на їхню думку, перебувають за межею можливостей і штучно ускладнені. Команда проекту приймає це до відома – простіше, звичайно, не буде, але способи викладання матеріалу і

тести постійно переглядаються, щоб якомога більше студентів змогли освоїти предмет, незважаючи на складність.

1.1.2 Ресурс Coursera для онлайн навчання

Coursera – це онлайн-платформа для навчання, заснована професорами Стенфордського університету Ендрю Нг і Дафні Коллером, яка пропонує курси, спеціалізації та ступені (рис.1.2). Coursera співпрацює з університетами та іншими організаціями з надання онлайн-курсів, спеціалізацій та ступенів з різних предметів, таких як інженерія, гуманітарні науки, медицина, біологія, соціальні науки, математика, бізнес, інформатика, цифровий маркетинг, інформатика та інші [2].

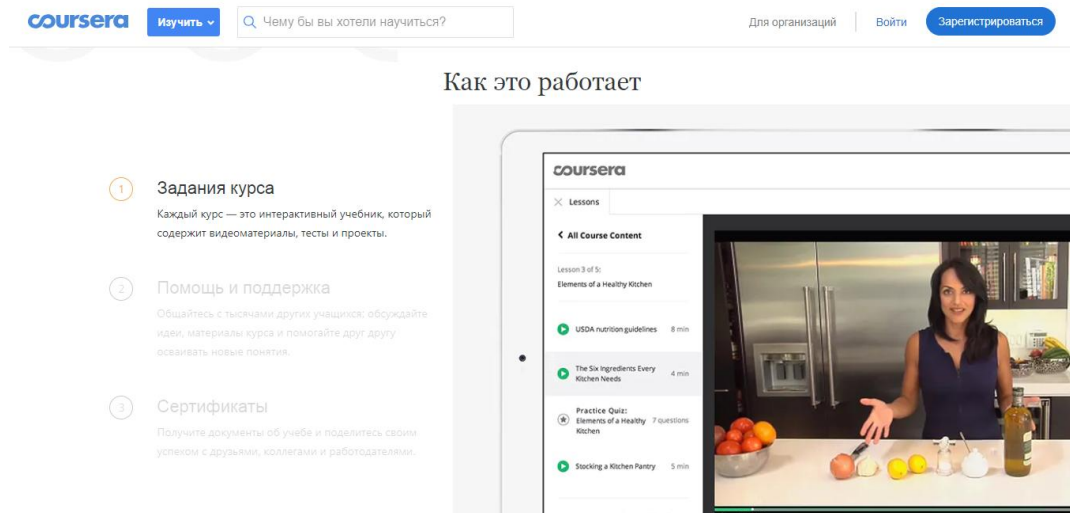


Рисунок 1.2 – Головна сторінка ресурсу Coursera

Coursera була заснована в 2012 році професорами інформатики Стенфордського університету Ендрю Нг та Дафні Коллером. Н.Г. і Коллер були натхнені своїм досвідом, пропонуючи свої курси Стенфорда в Інтернеті восени 2011 року, і незабаром після того, як він залишив Стенфорд, щоб запустити "Курсера". Принстон, Стенфорд, Мічиганський університет та Пенсильванський університет були першими університетами, які пропонують вміст

на платформі. Пропозиції з моменту розширення включають спеціалізації – збірки курсів, які складають навички у конкретній темі, а також ступені та продукти розробки робочої сили для бізнесу та державних організацій.

1.1.3 Ресурс Khan Academy для онлайн навчання

Khan Academy, відомий освітній ресурс, який надає можливість проходити курси платформи, де завгодно (рис. 1.3).





Math	Science & engineering	Computing	Arts & humanities	Economics & finance	Test prep	College, careers, & more
 Math		Early math Algebra 1 Trigonometry AP Calculus AB Differential equations	Arithmetic Geometry Precalculus AP Calculus BC Linear algebra		Pre-algebra Algebra 2 Statistics & probability Multivariable calculus	
 Math by grade		Kindergarten 3rd grade 6th grade Illustrative Mathematics	1st grade 4th grade 7th grade Eureka Math/EngageNY		2nd grade 5th grade 8th grade High school	
 Science & engineering		Physics Cosmology & astronomy Organic chemistry AP Biology	AP Physics 1 Chemistry Biology Health & medicine		AP Physics 2 AP Chemistry High school biology Electrical engineering	
 Computing		Computer programming Computer animation	Computer science		Hour of Code	

Рисунок 1.3 – Головна сторінка ресурсу Хан Академії

Khan Academy є неприбутковою освітньою організацією, створеною в 2008 році Салманом Хан, з метою створення набору Інтернет-інструментів, які допомагають навчати студентів. Організація випускає короткі уроки у формі відео YouTube. Її веб-сайт також включає в себе додаткові практичні заняття та матеріали для вчителів. Всі ресурси доступні для користувачів веб-сайту. Веб-сайт та його вміст надаються переважно англійською мовою, але також доступні іншими мовами, включаючи іспанську, португальську, іврит, італійську, російську, китайську, турецьку, французьку, бенгальську, хінді та німецьку мови [3].

1.1.4 Ресурс EdX для онлайн навчання

EdX – майданчик відкритої освіти, що передбачає безкоштовні онлайн-курси кращих університетів світу (рис.1.4). Він проводить онлайнві курси вищих навчальних закладів у широкому колі навчальних дисциплін для студентів у всьому світі, включаючи курси безплатно. Він також проводить дослідження в галузі навчання на основі того, як люди використовують свою платформу. EdX є неприбутковою організацією та працює на вільній програмній платформі Open open source openX [4].

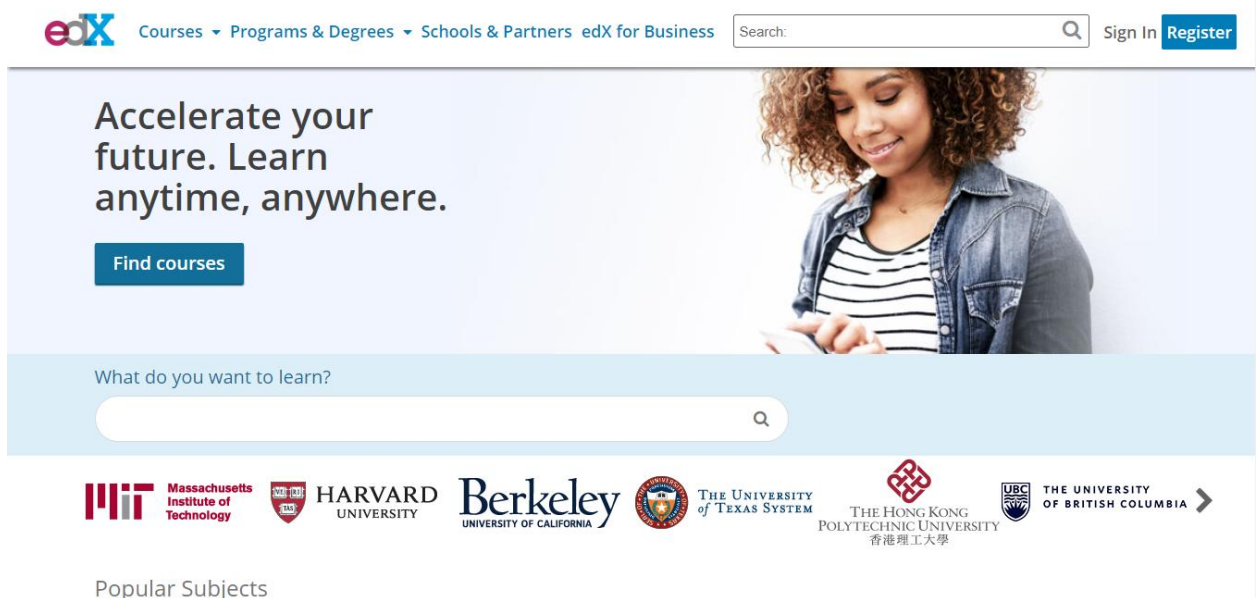


Рисунок 1.4 – Головна сторінка ресурсу EdX

Курси EdX складаються з щотижневих навчальних послідовностей. Кожна послідовність навчання складається з коротких відеороликів, що містяться в інтерактивному навчанні, де студенти можуть негайно практикувати концепції з відео. Курси часто включають навчальні відеоролики, схожі на невеликі дискусійні групи на навчальному закладі, онлайн-підручник та форум для обговорення в Інтернеті, де студенти можуть публікувати та переглядати питання та коментарі один одному та навчальні асистенти. У разі необхідності, онлайн-лабораторії включені в курс. Наприклад, в першому

МООС EdX – схемах та курсах електроніки – студенти створили віртуальні схеми в онлайн-лабораторії.

1.1.5 Порівняльний аналіз функціоналу систем аналогів

В рамках проведеного дослідження різних систем для навчання, складено таблицю з найважливішим функціоналом для будь-якої інформаційної системи навчання мов програмування.

Результати порівняльного аналізу функціоналу сучасних освітніх ресурсів аналогів наведені в табл. 1.1.

Таблиця 1.1 – Порівняльний аналіз функціоналу сучасних систем аналогів

Система	Udacity	EdX	Khan	Coursera	Власна розробка
Сертифікування	+	–	+	–	–
Тестування	+	–	–	–	+
Теорія	+	+	–	+	+
Дизайн	+	+	–	+	+
Можливість програмування в реальному часі	+	–	+	+	+
БД	+	–	+	+	+
Офлайн робота	–	–	–	+	+

Таким чином, створювана система містить в собі ідеї кожної з аналогічних, об'єднуючи в собі відкритість до покращення і додавання більш широкого функціоналу, але на відміну від існуючих аналогів є некомерційною та конкретно спрямованою на навчання програмуванню.

1.2 Вибір технологій для розробки системи

1.2.1 Вибір мови програмування

JavaScript – це мова програмування, що дозволяє зробити Web - сторінку інтерактивною, тобто такою що реагує на дії користувача [5].

JavaScript – об'єктно-орієнтована скриптова мова програмування і є діалектом мови ECMAScript. JavaScript зазвичай використовується як вбудована мова для програмного доступу до об'єктів додатків. Найбільш широке застосування знаходить у браузерах як мова сценаріїв для надання інтерактивності веб-сторінкам [5].

Основні архітектурні риси:

- динамічна типізація;
- автоматичне керування пам'яттю;
- прототипне програмування;
- функції як об'єкти першого класу.

JavaScript має низку властивостей об'єктно-орієнтованої мови, але реалізоване в мові прототипування обумовлює відмінності в роботі з об'єктами в порівнянні з традиційними об'єктно-орієнтованими мовами. Крім того, JavaScript має ряд властивостей, властивих функціональним мовам, - функції як об'єкти першого класу, об'єкти як списки, каррінг, анонімні функції, замикання – що додає мові додаткову гнучкість [6].

Синтаксис мови JavaScript дуже нагадує синтаксис C і Java, семантично ж мова набагато ближче до Self, Smalltalk або навіть Ліспу.

В JavaScript: всі ідентифікатори реєстрозалежні, в назвах змінних можна використовувати літери, підкреслення, символ долара, арабські цифри, назви змінних не можуть починатися з цифри, для оформлення однорядкових коментарів використовуються //, багаторядкові і внутрішньорядкові коментарі починаються з /* і закінчуються */.

Структурно JavaScript можна представити у вигляді об'єднання трьох частин, що чітко різняться одна від одної [7]: ядро (ECMAScript; об'єктна модель браузера (Browser Object Model або BOM); об'єктна модель документа (Document Object Model або DOM).

Об'єктну модель документа іноді розглядають як окрему від JavaScript сутність, що узгоджується з визначенням DOM як незалежного від мови інтерфейсу документа [5].

JavaScript класифікують як прототипну (підмножина об'єктно-орієнтованої), скриптову мову програмування з динамічною типізацією.

Окрім прототипної, JavaScript також частково підтримує інші парадигми програмування (імперативну та частково функціональну) і деякі відповідні архітектурні властивості.

Зокрема: динамічна та слабка типізація, автоматичне керування пам'яттю, прототипне наслідування, функції як об'єкти першого класу. Мова JavaScript використовується для:

- написання сценаріїв веб-сторінок для надання їм інтерактивності;
- програмування на стороні сервера (Node.js); стаціонарних застосунків (Electron, NW.js);
- мобільних застосунків (React Native, Cordova);
- сценаріїв в прикладному ПЗ (наприклад, в програмах зі складу Adobe Creative Suite чи Apache JMeter);
- всередині PDF-документів тощо.

1.2.2 Вибір та обґрунтування бази даних

Існують два найбільш поширених типу систем управління даними: реляційні БД (РБД) [12, 13] та NoSQL БД, різні в багатьох аспектах роботи.

В останні роки було розроблено велику кількість нових БД, які надають гарне горизонтальне масштабування для простих операцій читання, запису

для БД, розподілених на безлічі серверів, орієнтованих на Big Data [7]. На відміну від них, традиційні БД дають менше можливостей до масштабування.

Безліч нових БД визначаються терміном «NoSQL». Термін «NoSQL», що розшифровується як «Not Only SQL» або «Not Relational». Зазвичай такі системи задовольняють такими ознаками: наявність засобів розподілу навантаження на безліч серверів; можливість розподілу даних на безліч серверів; простий протокол виклику операцій; більш слабка модель паралелізму, ніж ACID-транзакції; ефективне використання розподілених індексів й оперативної пам'яті для зберігання даних; відсутність фіксованої схеми даних.

NoSQL-системи зазвичай не задовольняють властивостям ACID-транзакцій: допускається узгодженість в кінцевому рахунку. Пропонується модель «BASE» (Basically Available, Soft state, Eventually consistent, узгодженість в кінцевому рахунку) в протилежність ACID.

Ідея полягає в тому, що, відмовившись від обмежень ACID, можна домогтися набагато кращої продуктивності й масштабованості. Більшість систем різняться в ступенях відмови від ACID. Прихильники NoSQL часто посилаються на CAP-теорему, яка стверджує, що система може задовольняти лише двом з трьох таких властивостей: узгодженість - дані завжди однакові для всіх реплік; доступність - дані завжди доступні користувачеві; стійкість до поділу - система БД продовжує коректну роботу не дивлячись на відмову мережі або вузлів.

У NoSQL системах зазвичай опускається узгодженість, але припущення можуть бути складніше. Найчастіше моделі даних в NoSQL-системах розбиваються на наступні категорії [7]: сховища типу «ключ-значення» (Key-value stores): зберігають значення й ідентифікатор для пошуку, заснований на заданому ключі; документно-орієнтовані сховища (Document stores): система зберігає дані в формі документів, які індексуються; сховища, що розширюються записом (Extensible records stores): записи в таких сховищах можуть бути розподілені вертикально та горизонтально по вузлах.

Для середнього обсягу даних РБД та NoSQL дають схожі часові відгуки. Розбіжності проявляються при аналізі основних операцій.

Операції запису, оновлення, видалення даних потребують найбільшого часу на виконання для всіх типів БД. Найбільший час виконання операції запису, оновлення записів у DynamoDB при всіх обсягах даних. При середньому обсязі даних PostgreSQL та MySQL показує найкращий час виконання, ніж Cassandra, MongoDB. На великому обсязі даних швидше працює MongoDB.

Операції читання записів в БД PostgreSQL та MySQL виконуються швидше ніж NoSQL системах для середнього обсягу даних. DynamoDB значно програє в продуктивності ніж Cassandra, MongoDB та показує рівний результат при великому обсязі даних. Cassandra, MongoDB працюють значно швидше при великому обсязі даних для простих операцій читання [8].

Операція поновлення записів. Згідно результатів тестування продуктивність БД DynamoDB нижче інших систем. Найкращий показник продуктивності має БД MongoDB та Cassandra для великого обсягу даних. Операції Update в БД PostgreSQL та MySQL виконуються швидше ніж в NoSQL системах для середнього обсягу даних.

Операції підрахунку з угрупованням. У зв'язку з тим, що дана операція виконується в PostgreSQL, MySQL на стороні сервера, а для NoSQL вона реалізована на стороні користувача. PostgreSQL демонструє кращу продуктивність, ніж NoSQL системи. В свою чергу, продуктивність Cassandra, MongoDB вище, ніж продуктивність DynamoDB [9].

1.2.3 Опис стеку технологій

При розробці усіх модулів ресурсу були використані такі технології:

– мова програмування Python [10], яка була обрана для використання через швидкість написання програмного коду та його зрозумілість при подальшій роботі, а також через значну кількість доступних до використання біб-

ліотек, які реалізують різноманітну функціональність та позбавляють користувачів від необхідності створення власних програмних реалізацій;

- фреймворк Django [11], який дозволяє проводити створення веб-сайтів найбільш ефективно та швидко за рахунок можливості одразу виконувати додавання контенту на сайт за допомогою автоматично генерованої панелі адміністратора, що дозволяє з самого початку розробки побачити як контент відображається на кожній сторінці сайту;

- технологія віртуалізації Docker [12], яка дозволяє виконувати симуляцію незв'язності серверів для різноманітних СУБД та головного серверу ресурсу в окремих контейнерах, які у подальшому можна використовувати на справжніх серверах. Розробка модулю користувальницького інтерфейсу проводилась з використанням наступних технологій:

- мова гіпертекстової розмітки HTML [13], яка використовується для надання загальної структури веб-сторінок ресурсу;

- мова каскадних стилів CSS [14], яка використовується для визначення стилів різних елементів структури веб-сторінок;

- мова програмування JavaScript [6], яка використовується для реалізації частини функціоналу ресурсу, наприклад, використання технології AJAX для уникнення необхідності перезавантаження сторінки при незначній зміні даних. Розробка модулю контролеру, який забезпечує взаємодію з усіма іншими компонентами ресурсу, проводилась за допомогою:

- модуль мови програмування Python `django_postgres` [15], який використовувався для доступу до даних у PostgreSQL та маніпуляції з ними, забезпечення найбільшої зручності використання разом з фреймворком Django;

- модуль Python `elasticsearch_dsl` [15], використання якого дало можливість доступу до даних у Elasticsearch та базової маніпуляції з ними; – модуль мови програмування Python `django_redis` [15], який використовувався для доступу до кешу сторінок у Redis та для маніпуляції з ним, а також забезпечення найбільшої зручності використання разом з фреймворком Django;

- модуль мови програмування Python redis [16], який використовувався окремо від фреймворку Django для можливості самостійного доступу та маніпуляції з даними у базі;

- бібліотека мови програмування Python Scrapy [17], яка використовувалася для створення парсеру вакансій за рахунок максимальної зручності та простоти у використанні. Розробка модулів зберігання даних проводилась з використанням наступних СУБД:

- СУБД PostgreSQL [18], яка використовувалась для зберігання пов'язаних між собою даних та даних, які потребують найбільшої безпеки при збереженні та маніпуляції;

- СУБД Elasticsearch [19], яка використовувалась для забезпечення повнотекстового пошуку за словами запиту через значно більшу швидкість роботи цієї функції, ніж у інших СУБД;

- СУБД Redis [20], яка використовувалась для зберігання кешу сторінок та даних, які потребувалися лише тимчасово або була важлива їх швидка обробка за рахунок збереження усієї інформації у оперативну пам'ять.

2 ПРОЄКТУВАННЯ ОСВІТНЬОГО РЕСУРСУ

2.1 Мета та задачі інформаційної системи

Мета інформаційної системи: надання користувачам можливості проходження курсів та інших видів навчання у сфері інформаційних технологій для покращення свої знань або набуття нових; зробити навчання якомога більш доступним та швидким для широкого колу людей.

Продукт призначається для всіх, хто бажає отримати нові знання в сфері ІТ або зміцнити їх, для тих, у кого є бажання вивчати щось нове, але немає грошей на платні курси або недостатньо часу для їх відвідування.

Веб-сайт є безкоштовним освітнім ресурсом для навчання різним мовам програмування і технологій, що використовуються при розробці. На ресурсі присутні курси для початківців, тих, хто має деякі знання мов програмування та розробки в цілому, і розробників з достатнім досвідом роботи. Навчання проводиться в ігровій системі, що істотно збільшує інтерес користувачів до навчання. Інформація подається за допомогою відео, після кожного уроку користувач проходить тест, після проходження якого, можна говорити про засвоєння даної інформації. Для успішного завершення курсу, необхідно пройти модуль, який відкривається після проходження всіх уроків.

Важливою частиною функціонування освітнього ресурсу є роздільне зберігання використовуваної на ресурсі інформації в різних базах даних, які дозволяють краще і швидше працювати саме з певною інформацією. Це істотно прискорює роботу ресурсу і дозволяє робити завантаження сторінки більш швидкої, що сприятливо впливає на бажання користувача займатися навчанням саме в розроблюваному ресурсі.

Отже можна виділити наступні задачі системи:

- навчання користувачів за допомогою курсів за різними спеціальностями та рівнями підготовки;

- надавання додаткових джерел інформації у вигляді вебінарів та можливість поставити питання у коментаріях;
- перевірка знань користувачів після кожного уроку для визначення засвоєності навчального предмету;
- збільшення швидкості завантаження сторінок ресурсу та загальної швидкості його роботи за рахунок використання багатоваріантної персистенції.

Вхідні потоки: запити користувача до ресурсу.

Вихідні потоки: корисна для людини інформація у відповідь на запити.

На рис. 2.1 представлена контекстна діаграма, на якій відображені вхідні і вихідні потоки освітнього ресурсу.

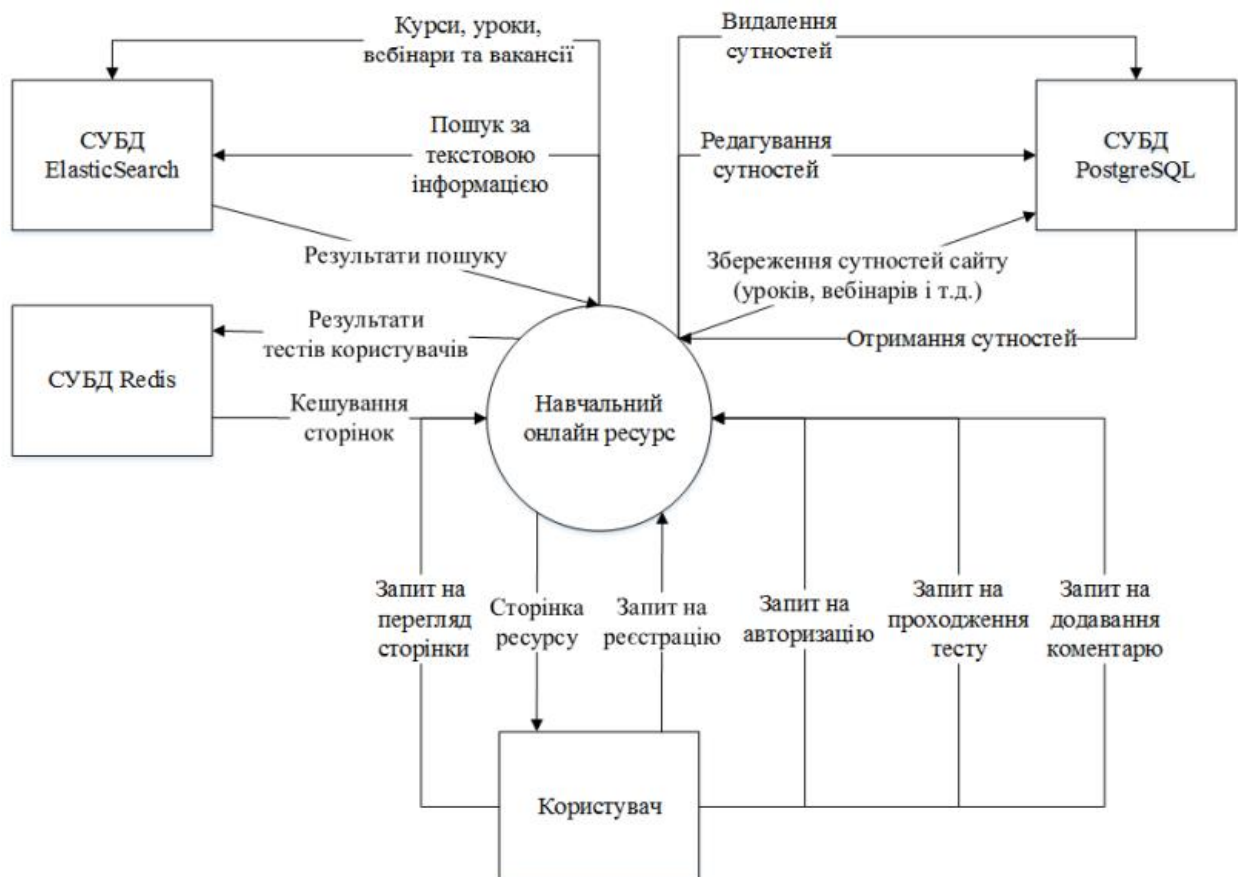


Рисунок 2.1 – Контекстна діаграма вхідних та вихідних потоків системи

До зовнішніх сутностей, які взаємодіють з системою, можна віднести користувача, який виконує операції перегляду сторінок, реєстрації, авторизації, редагування профілю, додавання коментарів, проходження тестувань і інше. Різні СУБД можна назвати зовнішніми, так як вони можуть знаходитися в різних локаціях, в тому числі і віддалених від самих серверів ресурсу, але при цьому функціонувати досить добре для виконання всіх функцій системи.

2.2 Керування вимогами

2.2.1 Типи користувачів

Гість – початковий користувач, який ще не пройшов реєстрацію або ще не увійшов до свого акаунту. Може відвідувати такі сторінки ресурсу: головна, список курсів, список вебінарів, список вакансій, сторінка інформації про ресурс, сторінка реєстрації, сторінка входу до облікового запису.

Звичайний користувач – користувач, який пройшов реєстрацію і увійшов до свого акаунту. Додатково до можливостей, які надаються гостю, він також може заходити на такі сторінки: сторінки окремих курсів і уроків, тестування після кожного уроку і проходження модуля, сторінку певного вебінару після реєстрації на нього, профіль і його редагування, а також у нього є можливість залишати коментарі до уроків і вебінарів.

Адміністратор має всі можливості звичайного користувача, а також може заходити на сторінку адміністрування, де має можливість видаляти будь-які з доданих елементів в базі даних, тобто курси, вебінари, спеціальності, коментарі і т.д.

2.2.2 Функціональні вимоги

FR1. Реєстрація / Авторизація.

FR1.1. Користувач повинен мати можливість зареєструватися на ресурсі.

FR1.2. Користувач повинен мати можливість авторизуватися на ресурсі, при цьому йому повинен бути доступний весь функціонал ресурсу.

FR1.3. Користувач повинен мати можливість вийти з поточного облікового запису.

FR1.4. Користувач-гість повинен мати можливість доступу до мінімуму функціональності ресурсу, а саме перегляду загальної інформації про курси, вебінари, вакансії і загалом про сайт.

FR1.5. Користувач повинен мати можливість виконати підписку на вебінар з отриманням повної функціональності сторінки вебінару, тобто можливості дивитися вебінар, залишати коментарі та переглядати коментарі інших учасників вебінару.

FR2. Дії з профілем.

FR2.1. Користувач повинен мати можливість перегляду профілю, в якому вказана інформація про нього, дана їм при реєстрації.

FR2.2. Користувач повинен мати можливість редагування інформації профілю і всіх полів, крім унікального, за яким здійснюється авторизація.

FR2.3. Користувач повинен мати можливість проглядання свого прогресу у вигляді заповненої шкали прогресу у відсотковому відношенні кількості пройдених уроків до кількості усіх уроків по кожному з курсів, які він проходить.

FR3. Пошук інформації.

FR3.1. Користувач повинен мати можливість зробити пошук за текстовою інформацією про курси (характеристикою курсу та назвою) та уроки (описом та назвою).

FR3.2. Користувач повинен мати можливість зробити пошук за текстовою інформацією про вебінари (описом, ключовими словами та назвою).

FR3.3. Користувач повинен мати можливість зробити пошук за текстовою інформацією про робочі вакансії (місту, у якому виконується пошук претендентів, вимогам та описом вакансії).

FR4. Коментування.

FR4.1. Користувач повинен мати можливість залишити коментар після ділянки з відео для кожного з уроків курсу.

FR4.2. Користувач повинен мати можливість побачити усі коментарі до всіх уроків кожного з курсів від інших користувачів.

FR4.3. Користувач повинен мати можливість залишити коментар після ділянки з відео вебінару, на який він підписаний.

FR4.4. Користувач повинен мати можливість побачити усі коментарі інших користувачів до вебінару, на який він підписаний.

FR5. Тестування для користувачів.

FR5.1. Авторизований користувач повинен мати можливість отримання доступу до тестів після кожного з уроків і проходження їх (надавання відповіді на тести у письмовій формі або вибору одного з варіантів у залежності від типу тесту), якщо він авторизований у системі.

FR5.2. Користувач повинен мати можливість проходження модуля по закінченню курсу для повного проходження курсу, якщо він авторизований у системі.

Виходячи з функціональних вимог було визначено діаграми прецедентів ресурсу. На рис. 2.2 представлена діаграма прецедентів для актора «Гість».



Рисунок 2.2 – Діаграма прецедентів для актора «Гість»

Дії з ресурсом, можливі до виконання після авторизації до акаунту були виділені в окрему частину діаграми для забезпечення можливості їх подальшого використання у декількох інших діаграмах без необхідності повторення (рис. 2.3).



Рисунок 2.3 – Прецеденти, які доступні для користувача після авторизації

На рис. 2.4 представлена діаграма варіантів використання ресурсу для актора «Звичайний користувач».



Рисунок 2.4 – Діаграма прецедентів для актора «Звичайний користувач»

На рис. 2.5 представлена діаграма прецедентів для «Адміністратора».

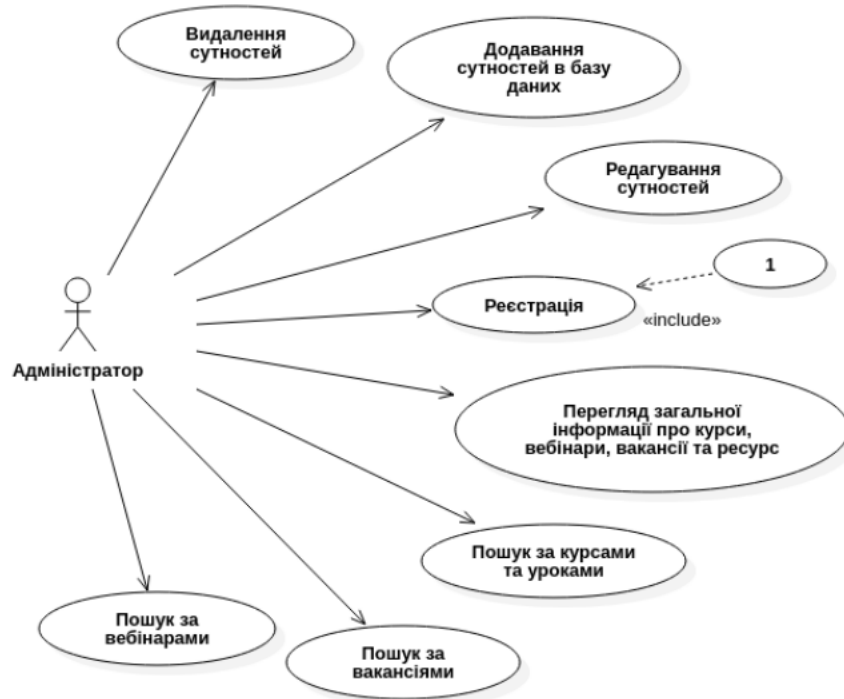


Рисунок 2.5 – Діаграма прецедентів для актора «Адміністратор»

З рисунку можна побачити, що адміністратору, окрім базової функціональності, а також повної функціональності, яка відкривається після авторизації, доступні також маніпуляції з усіма сутностями у ресурсі.

2.2.3 Нефункціональні вимоги

NFR1. Відповідь від сервера і завантаження сторінки повинно здійснюватися протягом не більше 3 секунд.

NFR2. Повинна забезпечуватися робота ресурсу в браузері Mozilla Firefox версії 5.0 ... 50.0.

NFR3. Повинна забезпечуватися робота ресурсу в браузері Opera версії 15.0 ... 57.0

NFR4. Повинна забезпечуватися робота ресурсу в браузері Google Chrome версії 5.0 ... 70.0.

2.3 Представлення користувальницького інтерфейсу

З будь-якої сторінки ресурсу можлива навігація по кнопках в головному меню, яке знаходиться вгорі кожної сторінки. Там присутні кнопки JustStart, яка виконує перехід на головну сторінку, Webinars (перехід на список вебінарів), Courses (перехід на список курсів), IT specialties (перехід на список вакансій), About us (перехід на інформацію про ресурс – про нас). Також правіше головного меню, в залежності від того авторизовані користувач, можуть відображатися або три кнопки (коли користувач авторизовані), зліва направо – перехід на профіль користувача, перехід в редагування профілю, вихід з облікового запису. Якщо користувач не авторизовані, замість трьох кнопок буде відображатися одна кнопка Login (перехід на сторінку авторизації – логін). На рис. А.1 представлена головна сторінка ресурсу.

На головній сторінці представлена базова інформація про сайт. Крім переходу по меню, на головній сторінці можна виконати перехід по кнопці Just Start в центрі екрану, яка переадресує користувача на сторінку реєстрації.

На сторінці реєстрації розгорнута форма реєстрування нового аккаунта користувача, яка зображена на рис. А.2. У формі присутні такі поля: ім'я користувача (*), адреса електронної пошти (*), посилання на Github аккаунт (*), країна проживання, пароль (*) і малюнок або фото, що представляє із себе характерне зображення користувача в профілі. Значком * відмічені обов'язкові поля у формі.

Якщо користувач натисне на кнопку Cancel, його буде переадресовано назад на головну сторінку. Якщо він заповнить всі необхідні поля правильно і натисне на кнопку Sign up, ресурс перенаправить його на сторінку профілю. Однак, якщо якісь поля будуть заповнені неправильно або ті поля, які є обов'язковими будуть незаповнені, йому буде видано попередження про це і буде очікуватися виправлення введеної інформації та повторна відправка форми реєстрації.

Сторінка логіна дозволяє здійснити вхід в акаунт (виконати аутентифікацію користувача) за допомогою імені користувача та пароля. Вона зображена на рис. А.3.

Кнопка Login переадресує користувача на сторінку профілю при успішній аутентифікації. Кнопка Sign Up перенаправляє на сторінку реєстрації, де користувач може створити новий обліковий запис. Кнопка Cancel повертає відвідувача на головну сторінку.

На сторінці "Про нас" представлена коротка інформація про ресурс, роботі сайту, а також цілі створення даного ресурсу. Ця сторінка зображена на рис. А.4. Навігація з неї можлива тільки по головному меню і посиланнях на соціальні мережі.

Сторінка списку з вакансіями (списку спеціальностей), яка зображена на рис. А.5, представляє з себе список з актуальними пропозиціями роботи в різних містах України в сфері ІТ. Актуальність досягається оновленням всього списку за допомогою створеного парсеру вакансій, який включається раз в день та повністю оновлює список вакансій.

Сторінка списку курсів містить всі наявні на сайті курси з різних мов програмування. Вона представлена на рис. А.6. Можна зробити пошук по курсам і уроків за допомогою поля для введення запиту. Сам пошук виконується після натиснення кнопки Search, яка зробить перенаправлення на таку ж сторінку зі знайденими курсами, які відповідають запиту. Пошук виконується за назвами, ключовими словами і описами уроків і курсів. Користувач перенаправляється на сторінку з вступом для курсу при натисканні на обраний курс.

На рис. А.7 представлена сторінка вступу курсу зі списком уроків. На початковій сторінці курсу знаходиться його короткий опис і посилання на уроки, а також посилання на модуль. Модуль доступний для проходження тільки після успішного виконання всіх тестів уроків. Посилання перенаправляють на відповідні сторінки уроків.

Сторінка уроку складається з опису, посилання на відео, секції з коментарями, поля для додавання коментарів і посилання на тест. Вона представлена на рис. А.8. Також на сторінці присутні посилання на інші уроки. Для додавання коментарів необхідно ввести текст в поле введення і натиснути на кнопку Send. Кнопка Pass Test виконує пересилку користувача на сторінку тесту. Після кожного уроку на ресурсі присутній тест. Проходження кожного тесту безпосередньо впливає на рівень користувача і прогрес по курсу, який відображається в профілі в секції Skills.

Сторінка зі списком вебінарів, яка представлена на рис. А.9, дозволяє побачити всі доступні на сайті вебінари, а також провести пошук по ним. Пошук працює абсолютно таким же чином, як і для курсів. При натисканні на обраний вебінар, відкривається сторінка з введенням для вебінару (сторінкою підписки на вебінар).

Сторінка вступу вебінару, яка зображена на рис. А.10, складається з короткої інформації про нього, кількості підписаних на нього користувачів і кнопки Join з назвою самого вебінару, яка дозволяє підписатися на нього і перенаправляє на сторінку вебінару.

Зовнішній вигляд сторінки вебінару представлений на рис. А.11. На ній присутні: його короткий опис, відео вебінару, форма для додавання коментарів і список коментарів до поточного вебінару. Для додавання коментарів необхідно ввести текст в поле і натиснути на кнопку Send.

Сторінка з тестами представляє собою кілька тестів, зазвичай близько двох-трьох. Приклад сторінки з тестом зображений на рис. А.12. Кожен тест знаходиться на своєму слайді, перемикання між ними відбувається по кнопці Next. Якщо не було дано відповіді на поточний тест, перехід на наступний буде неможливий. Коли користувач пройде всі тести на сторінці, йому буде показана кнопка Pass Test, розташована на місці кнопки Next. Питання в тестах бувають двох видів – відкритого типу, в якому є поле для введення і необхідно дати письмову відповідь і тест з варіантами відповідей, де

потрібно вибрати один варіант з кількох можливих. Після виконання тесту користувача перенаправить на сторінку показу результату тесту.

Сторінка результату тесту дає користувачеві можливість побачити зміг він пройти поточний тест чи ні. Один з варіантів сторінки результату тестування зображений на рис. А.13. Тест вважається пройденим, якщо відповіді на всі питання були вірними, інакше тест не зараховується. Якщо тест виконаний правильно, до прогресу користувача за поточним курсом додається бали, як і до прогресу по набору рівня.

На сторінці профілю (рис. А.14) відображена інформація про введені ним дані при реєстрації, а також про курси, за якими він проходив тести, в секції Skills і відображається набраний їм рівень відповідно до рівню досвіду за проходження уроків і курсів.

Сторінка редагування профілю зображена на рис. А.15. На ній можна виконати зміну інформації про користувача, а саме тієї, що була введена їм при реєстрації. При натисканні кнопки Save відбувається збереження даних і перенаправлення користувача на сторінку профілю за умови, що всі дані, що змінюються були введені вірно. Кнопка Cancel також виконує перенаправлення на сторінку профілю, але без збереження введених даних.

Виходячи зі створених макетів сторінок освітнього ресурсу, була побудована діаграма переходів станів, що показує можливі переходи з одного стану в інший, тобто з однієї сторінки на іншу (рис. 2.6).

З представленої діаграми можна побачити, що взаємодія з ресурсом відбувається за допомогою головного меню. Також, через те, що навчальний ресурс є веб-сайтом, закінчення роботи з ним можливе з будь-якої сторінки

2.6 Логічне представлення системи

Логічне представлення навчального ресурсу, а саме його структурні елементи, зв'язок між ними та приклади типів даних, що зберігаються у різних моделях БД, представлені на рис. 2.7. Ділянка, що представлена на

діаграмі, як middleware є основним компонентом ресурсу, який забезпечує зв'язок між усіма іншими компонентами сайту та виконує керування за усією логікою та функціональністю системи. Саме в ньому виконується реалізація концепції багатоваріантної персистентності, яка є основою функціонування ресурсу та забезпечує збільшення швидкості його роботи

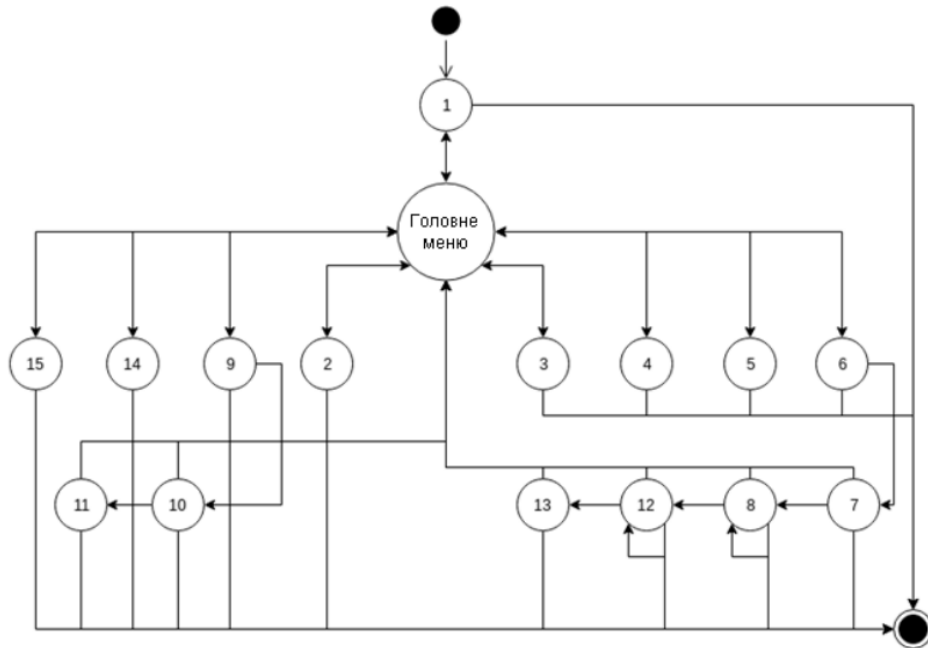


Рисунок 2.6 – Діаграма станів навчального ресурсу

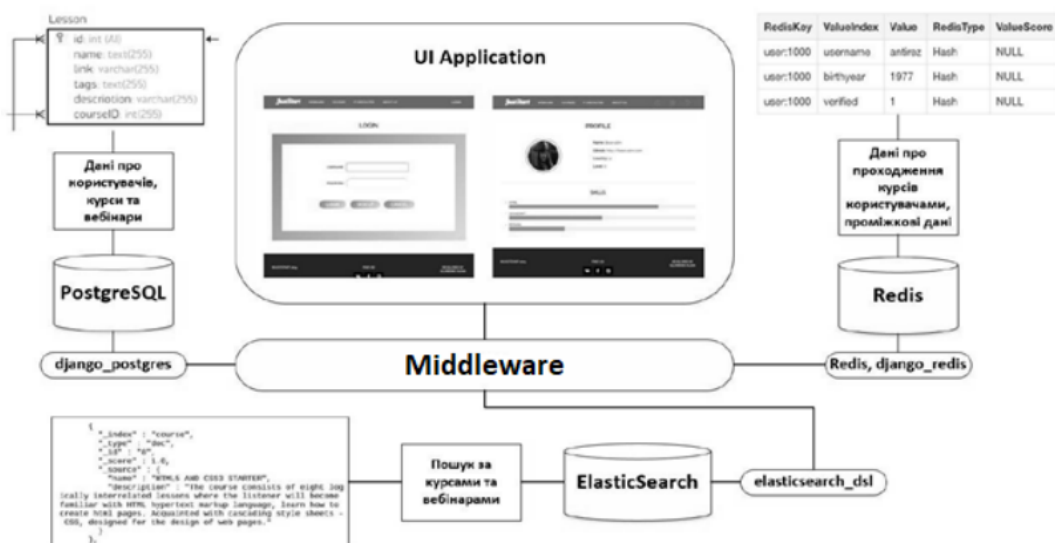


Рисунок 2.7 – Діаграма логічного представлення ресурсу

Графічний інтерфейс представляє собою інтерфейс для взаємодії з усім навчальним ресурсом та забезпечує доступ до всієї функціональності сайту, відповідно до типу користувача, який виконує взаємодію.

Middleware представляє собою головний сервер, який є проміжковим для усіх інших серверів для керування та передачі даних між ними. Керування здійснюється за допомогою взаємодії з різноманітними модулями роботи з БД для формування запитів або отримання даних, як результат цих запитів. При цьому формування запиту може бути не лише для отримання даних, але наприклад і редагування або видалення їх. Сервер працює у відповіді на дії користувача у графічному інтерфейсі.

Головний сервер частково використовує фреймворк `django` для взаємодії з базами даних. Робота з СУБД PostgreSQL відбувається за допомогою модуля `django_postgres` та використання об'єктно-реляційного відображення сутностей у базі та взаємодії з ними.

Робота з СУБД Redis виконується частково з використання цього фреймворку за допомогою модулю `django_redis` для мови програмування Python. Він дозволяє зберігати у цій БД кешоване представлення сторінок, які часто використовуються. Іншим використанням СУБД Redis є збереження проміжкової інформації, оновлення якої проводиться швидко або яка потрібна лише на деякий короткий час. Взаємодія з базою при такому сценарії ведеться за допомогою модулю `redis`.

Робота з СУБД Elasticsearch відбувається за допомогою модулю `elasticsearch_dsl` для забезпечення зберігання індексованих текстових даних, за якими користувач може виконувати пошук, та подальшого отримання даних, слова у яких співпадають зі словами пошукового запиту.

СУБД PostgreSQL використовується у навчальному ресурсі, як RDBMS – реляційна база даних, для збереження даних про користувачів, запис їх на курси та вебінари, зв'язки між курсами, уроками, тестуванням та іншими сутностями. Також у ньому зберігаються дані, для яких необхідно забезпечити найбільшу безпеку збереження та отримання.

СУБД Elasticsearch використовується у ресурсі для забезпечення пошуку за курсами, уроками та вебінарами за рахунок дуже швидкого доступу до текстових даних через використання хешованих значень кожного слова.

СУБД Redis використовується для збереження кешу сторінок та ресурсів, які змінюються не часто та не потребують виконання перезавантаження після оновлення сторінки. Це дозволяє значно збільшити швидкість завантаження сторінок та інших додаткових ресурсів. Також ця технологія використовується для збереження проміжкової інформації, наприклад для збереження відповідей користувача на тестування для перевірки їх на сервері та переадресації користувача на відповідну до результату сторінку.

2.7 Представлення розгортання ІС

Діаграма розгортання застосовується для уявлення загальної конфігурації і топології розподіленої програмної системи і містить зображення розміщення компонентів по окремих вузлах системи. Крім того, діаграма розгортання показує наявність фізичних з'єднань – маршрутів передачі інформації між апаратними пристроями, задіяними в реалізації системи.

ІС складається з веб-серверу, баз даних та комп'ютера користувача, що зображено на діаграмі представлення розгортання (рис 2.8).

Користувач переходить за адресою сайту, після чого взаємодіє з компонентами сайту. Для обробки дій користувача та передачі даних використовується протокол HTTP. Дані відправляються на сервер.

Сервер Django WSGI Server розташований у компоненті програмного забезпечення Docker. Django-проект має зв'язки з різними БД, такими як PostgreSQL, Elasticsearch, Redis.

Компонент PostgreSQL використовується для зберігання даних про користувачів, запис їх на курси та вебінари, зв'язки між курсами, уроками,

тестуванням та іншими сутностями. Компонент серверу взаємодіє з ним за допомогою бібліотеки `django_postgres`.

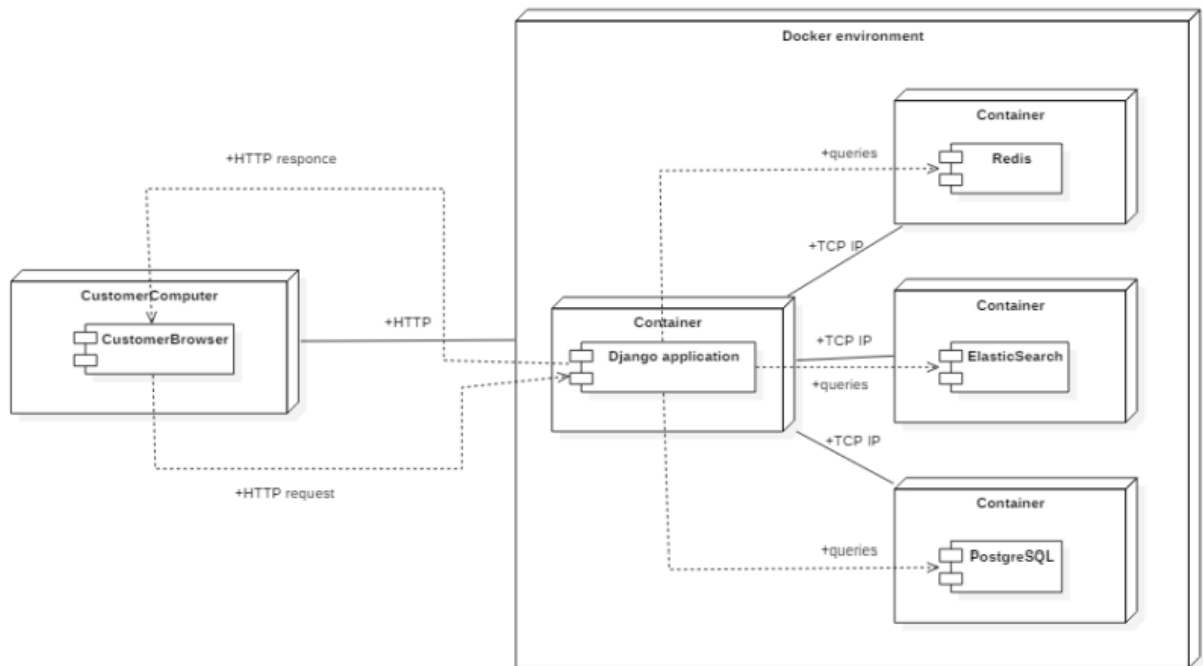


Рисунок 2.8 – Представлення розгортання ІС

Компонент Elasticsearch використовується для забезпечення пошуку за курсами, уроками та вебінарами за рахунок дуже швидкого доступу до текстових даних через використання хешованих значень кожного слова. Компонент серверу взаємодіє з ним за допомогою бібліотеки `elasticsearch_dsl`.

Компонент Redis використовується для збереження кешу сторінок та ресурсів, які змінюються не часто. Компонент серверу взаємодіє з ним за допомогою бібліотеки `django_redis`.

2.8 Представлення шарів ІС

Логічна група компонентів, які можуть бути повторно використані в подібних обставинах, і мають спільні цілі називається шаром. Компоненти

всередині одного шару пов'язані один з одним. На рис. 2.9 представлена діаграма шарів. Нижче описано компоненти, з яких складається ІС.

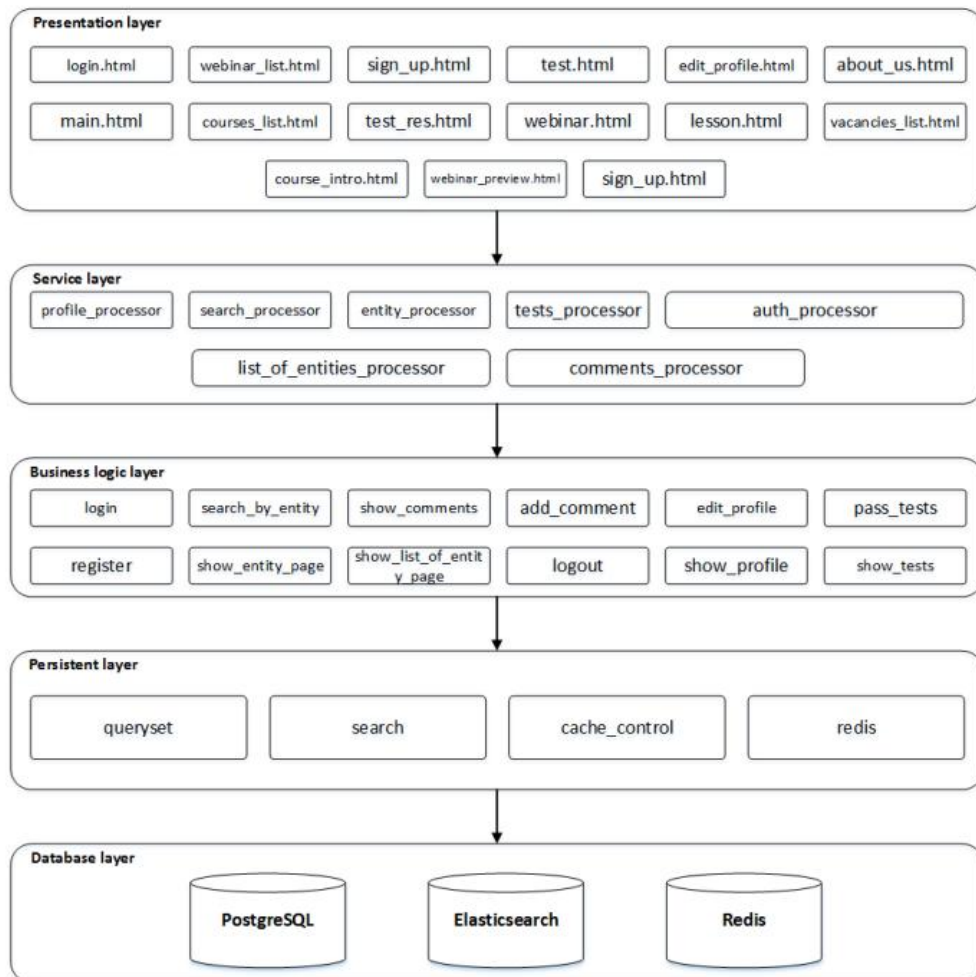


Рисунок 2.9 – Представлення шарів ІС

Рівень «Presentation layer» складається з компонентів, що відповідають за відображення елементів на екрані, введення даних та інші взаємодії з користувачем. Вони відповідають за вигляд користувацького інтерфейсу.

Рівень «Service layer» складається з компонентів, що відповідають за обробку дій користувача. Залежно від роду дії кожний компонент даного рівня контролює, делегує та розподіляє відповідальність між компонентами нижнього рівня.

Компонент «profile_processor» відповідний за взаємодію користувача з профілем.

Компонент «search_processor» містить сценарії стосовно пошуку курсів та вебінарів у онлайн навчальному ресурсі.

Компонент «list_of_entities_processor» відповідний за відображення та роботу списків з вебінарами або курсами.

Компонент «entity_processor» містить функціонал стосовно кожної окремої самостійної одиниці списку курсів або вебінарів.

Компонент «tests_processor» обробляє дії користувача під час тестування на сайті.

Компонент «auth_processor» відповідальний за авторизацію та реєстрацію користувачів у системі.

Компонент «comments_processor» містить логіку, що відповідає за взаємодію та відображення коментарів на онлайнному навчальному ресурсі.

Рівень «Business logic layer» містить загальну логіку онлайн навчального ресурсу. Кожний компонент даного рівня містить детальні інструкції щодо обробки вхідних даних та взаємодії з різними БД для збереження, редагування чи видалення інформації:

- компонент «login» відповідає за авторизацію користувача в системі онлайнного навчального ресурсу;

- компонент «search_by_entity» містить сценарій щодо пошуку по назві та вмісту у списках курсів або вебінарів;

- компонент «register» відповідає за реєстрацію в системі в разі, якщо такого користувача немає;

- компонент «show_entity_page» оперує функціоналом щодо відображення детальної інформації стосовно курсу або вебінару;

- компонент «show_comments» відповідний за вилучення коментарів з СУБД та їх відображення користувачеві у системі;

- компонент «show_list_of_entity_page» відповідальний за відображення списку з курсами або вебінарами користувачеві після їх вилучення з СУБД;
- компонент «add_comment» містить сценарій додавання нового коментаря користувачем та порядок його обробки для подальшого збереження у СУБД;
- компонент «logout» відповідає за вихід користувача з системи онлайн-навчального ресурсу;
- компонент «edit_profile» відповідальний за обробку та збереження в СУБД особистих даних користувача після їх редагування;
- компонент «show_profile» містить сценарій для вилучення та відображення даних профілю користувача на веб-сайті;
- компонент «pass_tests» відповідає за отримання даних, обробку відповідей та збереження прогресу користувача в СУБД;
- компонент «show_tests» відповідальний за вилучення питань та варіантів відповідей із СУБД та їх правильне відображення користувачеві.

Рівень «Persistent layer» складається з компонентів, що відповідають за роботу з різними СУБД. Дані компоненти є бібліотеками, що з'єднуються з проектом для автоматизації обробки «сирих» даних у зручний вид для подальшої роботи з ними, а також вилучення та збереження їх у відповідну СУБД.

Для роботи з базою даних Django використовує власний ORM, в якому модель даних описується класами Python, і по ній генерується схема бази даних.

ORM (об'єктно-реляційне відображення, або перетворення) - технологія програмування, яка зв'язує бази даних з концепціями об'єктно-орієнтованих мов програмування, створюючи «віртуальну об'єктну базу даних». Існують як пропрієтарні, так і вільні реалізації цієї технології.

Системи ORM знаходять все більшу популярність серед розробників, так як вони автоматизують безліч типових взаємодій з базою даних і

використовують знайомий об'єктно-орієнтований підхід замість інструкцій SQL:

- компонент «queryset» відповідальний за обробку основних операцій з БД, який використовує модуль для виконання низькорівневих запитів до СУБД; виконує повернення результатів компонентам інших рівнів з СУБД PostgreSQL;

- компонент «search» являє собою допоміжний двигун для простої реалізації швидкого пошуку у СУБД Elasticsearch;

- компонент «cache_control» забезпечує взаємодію компонентів інших рівнів з СУБД Redis для зберігання кешу сторінок;

- компонент «redis» забезпечує взаємодію компонентів інших рівнів з СУБД Redis для зберігання тимчасових даних або даних, які потребують швидкої обробки.

Рівень «Database layer» складається з компонентів, що являють собою СУБД, що використовуються для вирішення різних задач відповідно:

- компонент «PostgreSQL» оперує даними про користувачів, вебінари, курси та вакансії;

- компонент «ElasticSearch» індексує вміст PostgreSQL та забезпечує швидкий пошук по онлайн-навчальному ресурсі;

- компонент «Redis» являє собою СУБД, що відіграє головну роль у забезпеченні збереження кешу сторінок та проміжних даних.

2.9 Представлення процесів ІС

Веб-ресурс має декілька основних бізнес-сценаріїв – авторизація користувача, проходження курсів та тестування, перегляд вебінарів.

Сценарій авторизації зображено на рис. 2.10.

Валідація форми авторизації відбувається без необхідності відправки даних на сервер, що значно зменшує навантаження на нього.

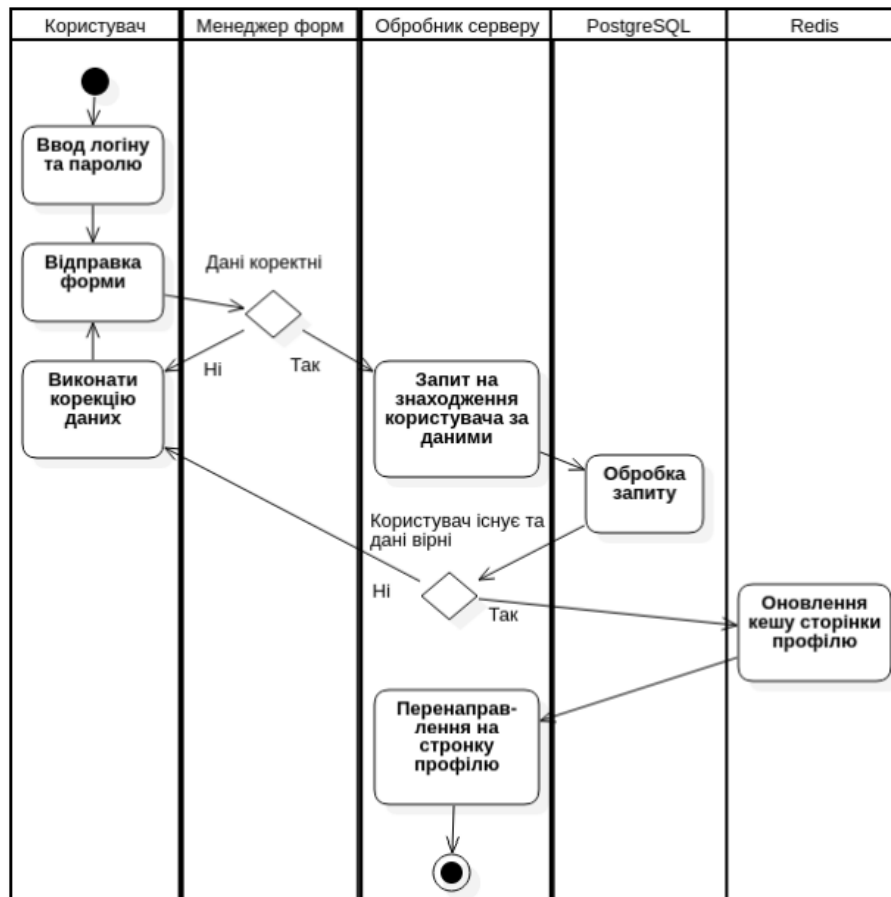


Рисунок 2.10 – Діаграма діяльності авторизації

При авторизації нового користувача відбувається примусове оновлення кешу сторінки профілю, який зберігається для збільшення швидкості взаємодії з ресурсом та зменшення навантаження на сервер. Його видалення необхідно, щоб до іншого користувача системою на цьому комп'ютері не потрапила зайва інформація попереднього користувача.

На рисунку 2.11 зображена взаємодія компонентів системи при авторизації.

Доступ до СУБД PostgreSQL забезпечується за допомогою спеціального модулю фреймворку Django – `queryset`, що забезпечує роботу з кожним з об'єктів ORM Django.

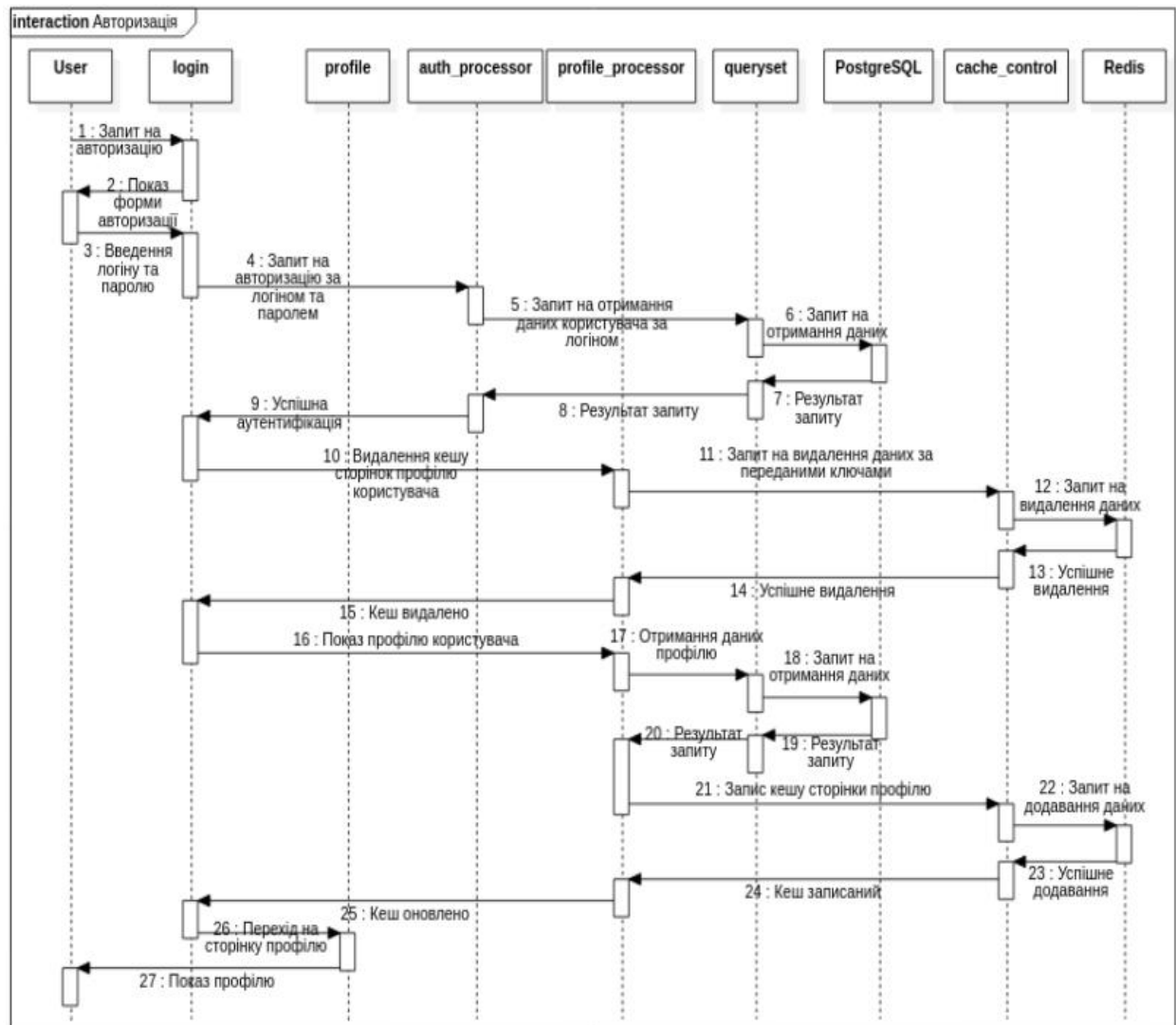


Рисунок 2.11 – Діаграма послідовності авторизації

Доступ до СУБД Redis забезпечується за допомогою модулю `cache_control`, який використовується для зберігання кешу сторінок та часто використовуваних ресурсів на сторінці.

На рис. 2.12 зображена діаграма діяльності пошуку користувачем за курсами.

Зберігання індексованих даних таблиць з СУБД PostgreSQL у СУБД ElasticSearch дає можливість значно збільшити швидкість пошуку за текстовою інформацією на відміну від використання SQL запитів.

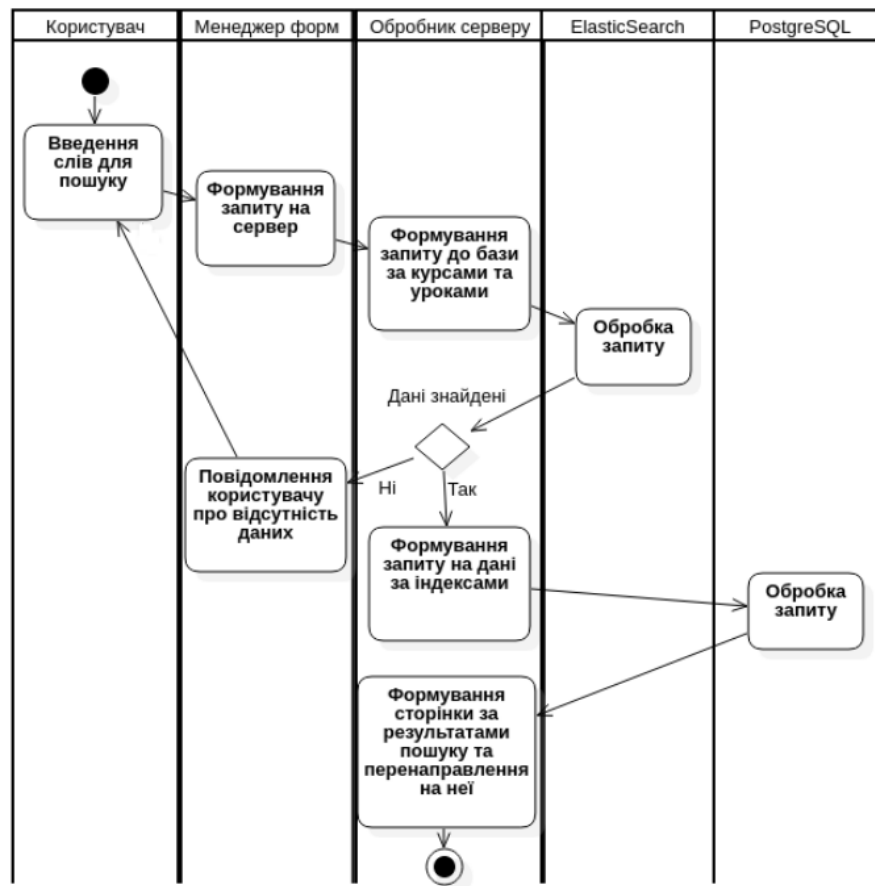


Рисунок 2.12 – Діаграма діяльності пошуку за курсами

Сторінки списків з даними були розроблені з загальним алгоритмом роботи: якщо користувач не ввів запит на пошук, то показуються усі існуючі дані, пов'язані з даною сторінкою (тобто курси, вебінари або інші); якщо користувач запровадив дані запиту, будуть показані лише ті дані, які мають спільні з запитом слова; якщо нічого не було знайдено за даним запитом, користувачу буде показане повідомлення про відсутність співпадань серед існуючих даних.

На рис. 2.13 представлена діаграма послідовності пошуку користувачем за курсами.

Доступ до СУБД ElasticSearch забезпечується за допомогою модулю search, який виконує інтеграцію з фреймворком Django та дає можливість

автоматично додавати дані до індексів разом зі створенням сутностей у СУБД PostgreSQL.

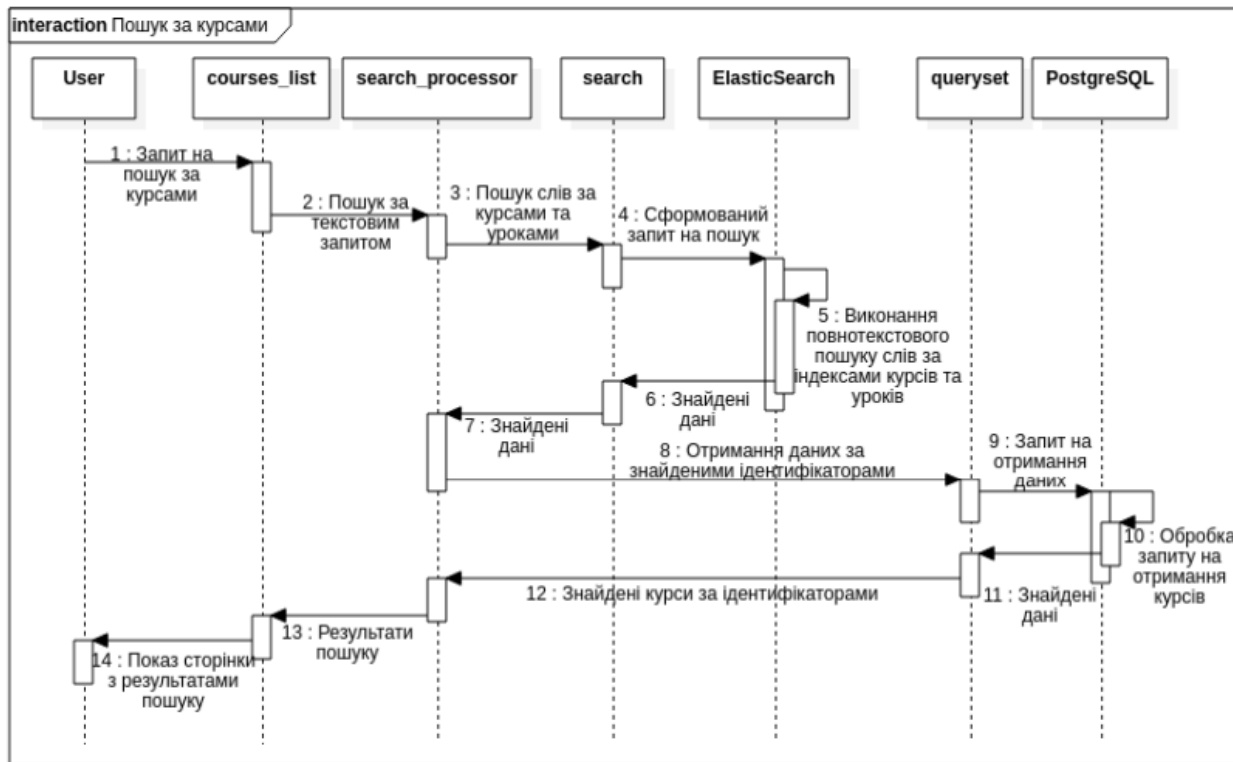


Рисунок 2.13 – Діаграма послідовності пошуку за курсами

На рис. 2.14 зображена діаграма діяльності виконання тестування за уроком деякого курсу. Тести завантажуються на сторінку один раз та відображаються один за одним після проходження попереднього.

Зазвичай кожен урок має два або три тести, на усі з яких користувач повинен відповісти правильно для збільшення прогресу за цим курсом та отримання підвищення опиту для збільшення ігрового рівня. Діаграма послідовності тестування за уроком певного курсу зображена на рис. 2.15.

Відповіді користувача на тест зберігаються тимчасово у СУБД Redis для подальшого порівняння з дійсними правильними відповідями на сервері. Такий підхід дозволяє зменшити кількість запитів до головної СУБД ресурсу.

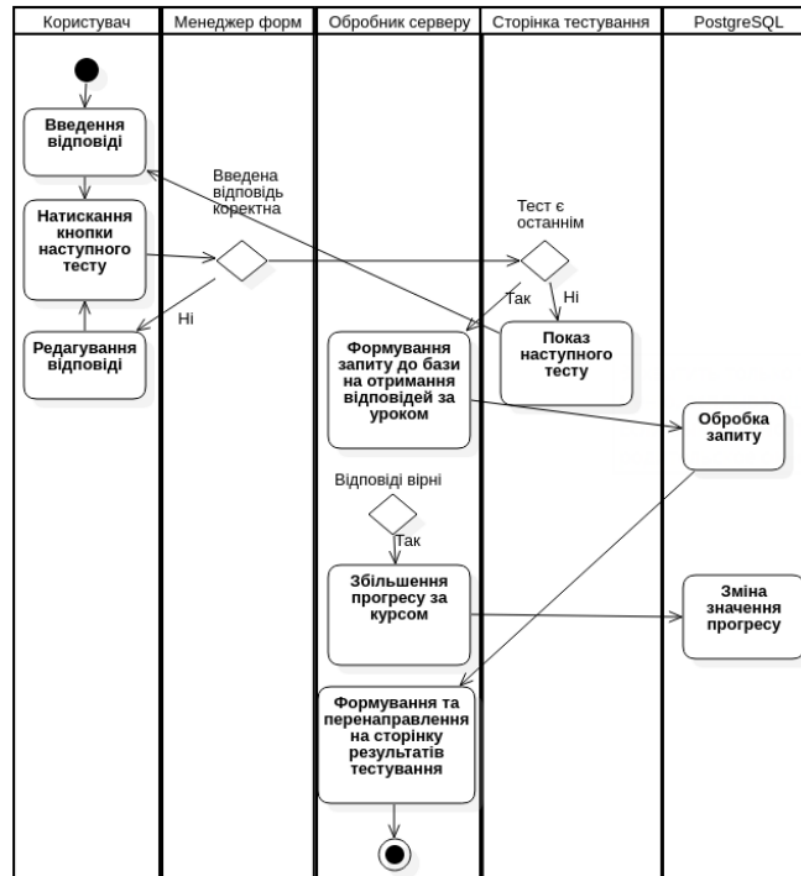


Рисунок 2.14 – Діаграма діяльності виконання тестування

Користувач повинен ввести відповідь на тест, якщо це тест відкритого типу або обрати один з варіантів відповідей, якщо це тест з декілька варіантами для переходу до іншого тесту. Відповіді користувача передаються на сервер для обробки лише після проходження усіх тестів.

2.10 Представлення даних ІС

Система потребує багато ресурсів та оперує великою кількістю різних за типом даних, тому для зменшення навантаження на сервер та збільшення швидкості завантаження сторінок, а також збільшення швидкості роботи ресурсу було використано багатоваріантну персистентність з використанням трьох баз даних, які функціонують окремо одна від одної.

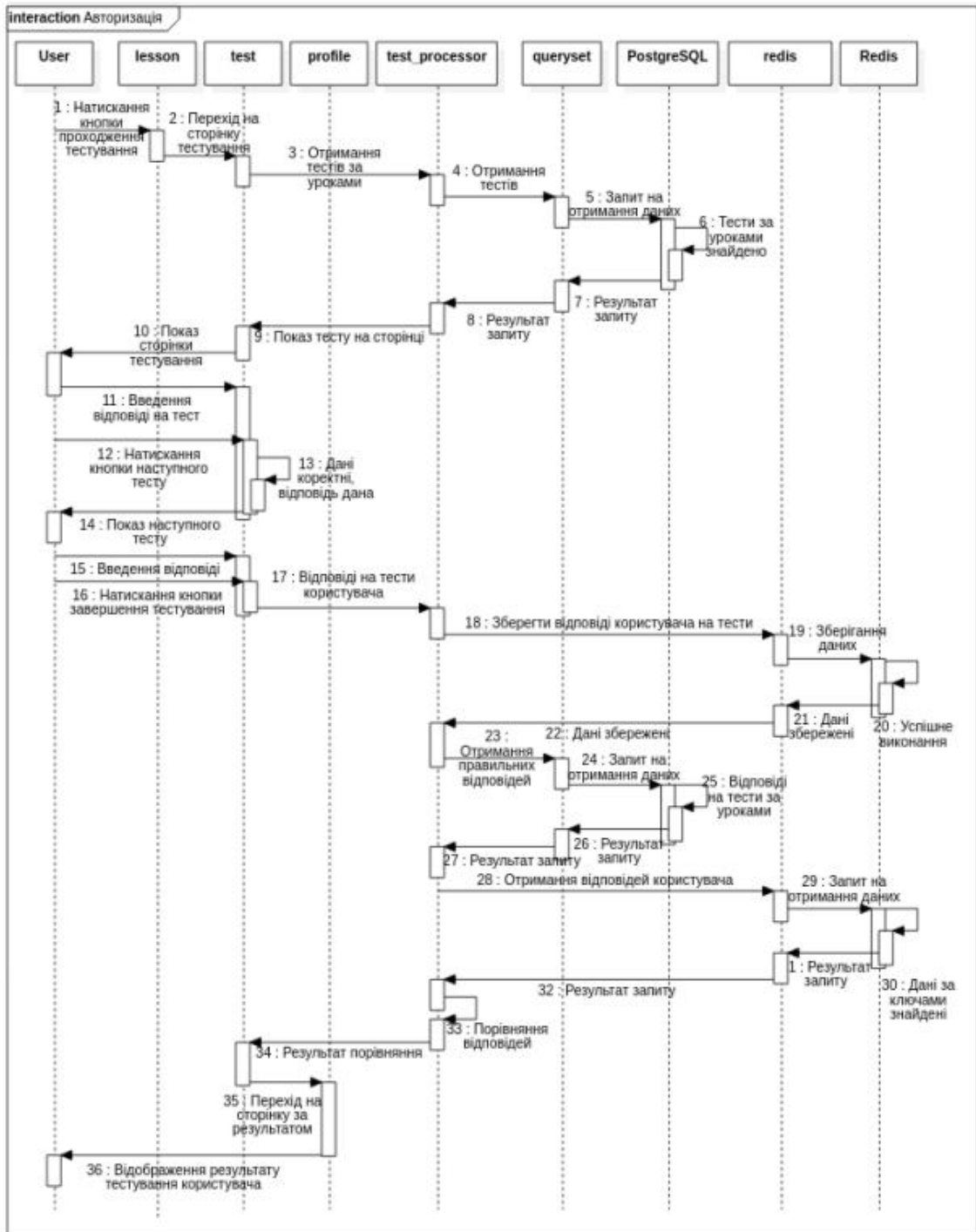


Рисунок 2.15 – Діаграма послідовності виконання тестування

За допомогою СУБД ElasticSearch у навчальному ресурсі проводиться текстовий пошук введених користувачем слів. Ця технологія не може забезпечити достатньої простоти для обробки зв'язних даних, що відбувається у SQL базах даних за рахунок побудови структури таблиць.

Тобто вона буде потребувати ручного видалення відповідних значень у базі даних, при видаленні пов'язаних з ним інших даних. Але при цьому за допомогою індексації, тобто збереження хешу від кожного зі слів тексту окремо, можна проводити пошук певних слів набагато швидше, ніж при використанні інших СУБД для цих цілей.

У СУБД Elasticsearch дані зберігаються у виді, так званих, індексів, які є аналогом таблиць у реляційних базах даних. Внутрішня структура індексів схожа за видом на тип даних «словник» у мові програмування Python, а також на тип json файлів. Тобто вона складається з деякої кількості об'єктів типу ключ-значення. На рис. 2.16 представлений приклад зовнішнього вигляду даних, що зберігається у СУБД Elasticsearch.

```
{
  "_index": "course",
  "_type": "doc",
  "_id": "12",
  "_score": 1.0,
  "_source": {
    "name": "JAVA ESSENTIAL VIDEO COURSE",
    "description": "Teaching programming takes place, starting with the study of the basic concepts and principles of object-oriented programming, classes, and also the features of the Java language, and ending with the consideration of the basic concepts of such a paradigm as Java Generics."
  }
},
{
  "_index": "lesson",
  "_type": "doc",
  "_id": "7",
  "_score": 1.0,
  "_source": {
    "name": "LISTS",
    "description": "In the lesson you will learn about the concept of a list, indices, list cross sections, checking elements for entry into the list, working with lists (getting the values of elements, adding and deleting values).",
    "tags": "The concept of lists and arrays. Lists in python. Creating Lists. Indexing lists. List slices. Operation in. Len function Strings. Examples. List operations: add, delete, edit items. List traversal. An example of faster calculation of Fibonacci numbers."
  }
}
```

Рисунок 2.16 – Представлення збережених даних у СУБД Elasticsearch

У навчальному ресурсі використовується чотири типа індексів для Elasticsearch для зберігання даних з таблиць курсів, уроків, вебінарів і вакансій, що зображено на рис. 2.17.

У курсах індексується назва і опис; в уроках – назва, опис і ключові слова; в вебінарах – теж назву і опис; у вакансіях – назва, місто, для якого виконується пошук, вимоги та опис.

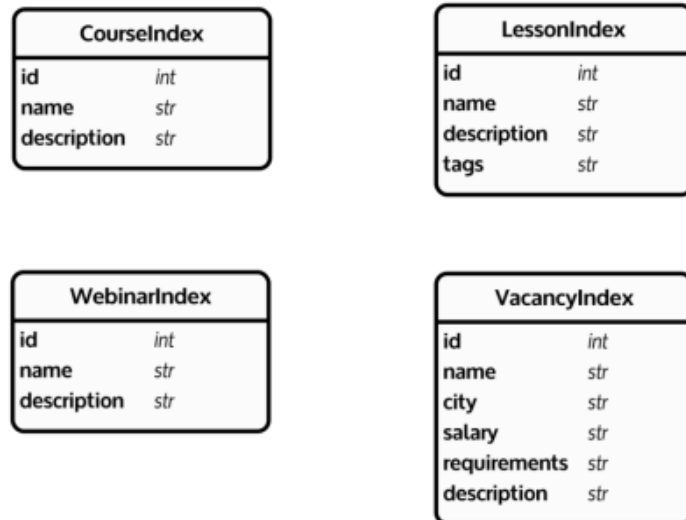


Рисунок 2.17 – Представлення індексів у СУБД Elasticsearch

СУБД Redis використовується у ресурсі для зберігання кеша даних сторінки користувача для прискорення доступу до неї, інших часто використовуваних сторінок, дані на яких змінюються досить нечасто, а також для тимчасового зберігання даних, які потребують швидкої обробки або потрібні лише на короткий проміжок часу, наприклад, тимчасове зберігання відповідей на питання при проходженні тесту користувачем для їх подальшої перевірки та визначенні чи пройшов користувач тестування.

Дані в цій СУБД мають формат пар ключ-значення. Приклад таких даних представлено на рис. 2.18. Доступ до них отримується дуже швидко, за рахунок того, що вони зберігаються в оперативній пам'яті пристрою користувача, але це значно обмежує використання цієї технології, бо величина цих даних не може перевищувати величину оперативної пам'яті пристрою.

У кожного файлу кеша є свій ключ, який вказує на нього. Кеш примусово оновлюється тільки в двох випадках: користувач або адміністратор редагував дані, які присутні на сторінці або, наприклад, користувач вийшов з облікового запису, якщо мова йде про кеш профілю

користувача. В такому випадку видаляються всі ключі зі старими даними на машині користувача і оновлюється кеш.

key	value
Alina	Programming language; Python code must be always compiled; print
AlinaGL	4;7; world
Val	74;3;/: 6;elif; world;3;3;r; Python code must be always compiled
Glum	3;6; shout!

Рисунок 2.18 – Представлення збережених даних у СУБД Redis

СУБД PostgreSQL є реляційною БД, що забезпечує гарну роботу з даними, пов'язаними між собою значним чином. Але при цьому ця технологія не є кращим варіантом для зберігання даних, які необхідно використовувати часто або виконувати швидку обробку великої кількості даних, через значне зростання необхідного для виконання обробки запиту часу, при збільшенні об'єму даних.

На рис. 2.19 зображено діаграму сутностей системи, що показує організацію та взаємодію сутностей, збережених у СУБД PostgreSQL. Ці сутності були створені за допомогою об'єкт-реляційної моделі фреймворку django.

Опис представлених в діаграмі сутностей:

– сутність Course є окремим курсом навчання, у якого є опис, який відображається у вступі до курсу і його назва, а також поле викладача. – сутність Lesson є окремим уроком курсу. Кожен урок має назву, опис, посилання на відео в уроці, ключові слова і посилання на курс, якому належить цей урок.

– сутність Webinar є вебінаром, що проводиться одним викладачем. У нього є назва, опис, поле викладача, посилання на відео вебінару, його тривалість і дата проведення.

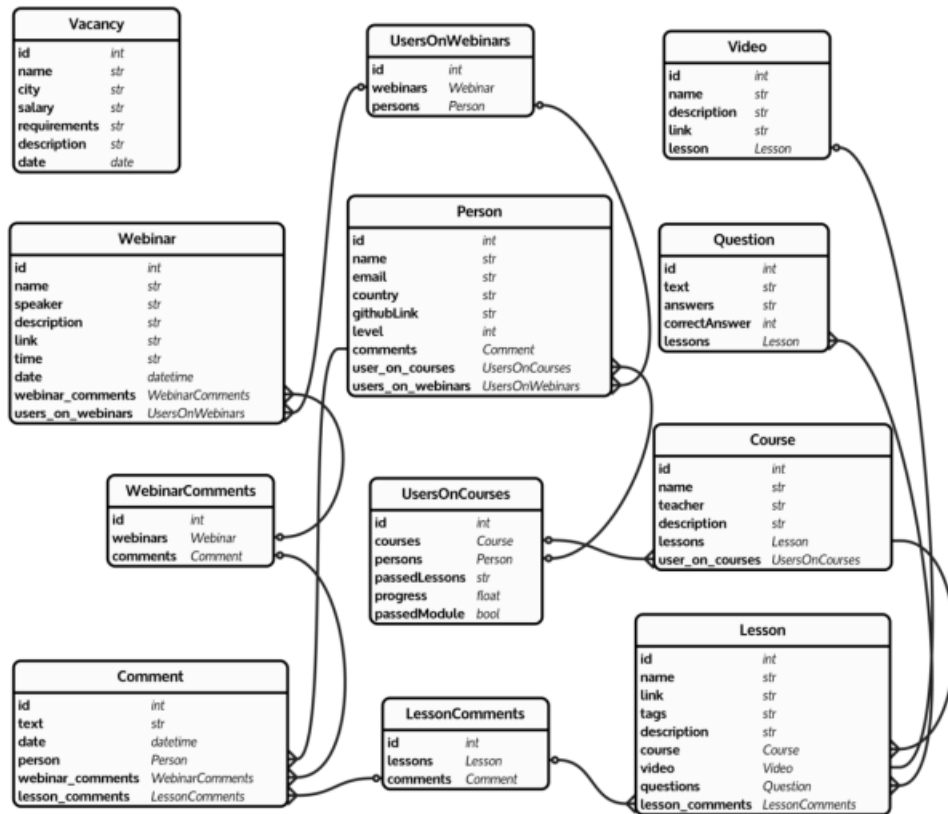


Рисунок 2.19 – Діаграма сутностей у СУБД PostgreSQL

– сутність *Vacancy* являє собою вакансію на роботу. У неї є такі поля: найменування необхідного фахівця, місто, для якого актуальна вакансія, пропонована заробітна плата, вимоги до фахівця, опис вакансії, дата додавання вакансії.

– сутність *Person* являє собою користувача ресурсу. У сутності є такі поля: ім'я користувача (логін), посилання на github аккаунт, країна проживання, пароль, адресу електронної пошти і рівень, якого досяг користувач, проходячи тести з курсів.

– сутність *Comment* являє собою коментар користувача, в якому є посилання на коментатора, текст коментаря і дата його додавання.

– сутність *Question* представляє собою окремий тест, який може бути двох видів – закритого (з декількома варіантами відповіді) і відкритого (з одним варіантом відповіді). У нього є такі поля: питання, відповідь

(правильна), і варіанти відповідей (або прочерк, якщо питання відкритого типу), а також посилання на урок, якому належить цей тест. – сутність WebinarComments являє собою зв'язок між окремим вебінаром і коментарями до нього.

– сутність LessonComments являє собою зв'язок між окремим уроком і коментарями до нього.

– сутність UsersOnCourses являє собою зв'язок між користувачами і курсами, які вони проходять. У кожного користувача є певний рівень прогресу по розпочатому їм курсу.

– сутність UsersOnWebinars являє собою зв'язок між користувачами і вебінаром, на які вони записані.

3 РЕАЛІЗАЦІЯ ОСВІТНЬОГО РЕСУРСУ

3.1 Уявлення про структуру проекту ІС

Розробка здійснювалася за спроектованими схемами та діаграмами, дотримуючись основних вимог до освітнього ресурсу та його швидкодії. Увага також приділялась особливостям використання та розробки на мові програмування Python, веб-фреймворку Django та використаним у кваліфікаційній роботі СУБД.

На рис. 3.1 зображена структура проекту, яка складається з базового пакету «composer» та пакету «education», який розширює його.



Рисунок 3.1 – Структура проекту ІС

Пакет «composer» містить файли «__init__.py», «settings.py», «urls.py», «wsgi.py». Файл «settings.py» містить налаштування проекту, файл «urls.py» -

базову адресацію за посиланнями, а саме перехід на головну сторінку та на сторінку адміністратора.

Пакет «education» являє собою модуль проекту, який містить файли «_init.py», у якому відбувається ініціалізація початкової конфігурації проекту, «admin.py», де відбувається підключення моделей проекту (тобто таблиць у СУБД PostgreSQL) до можливості додавання та редагування сутностей цих моделей у панелі адміністратора. Файл «forms.py», де визначені усі форми та їх поля, що використовуються у ресурсі. Файл «models.py», що визначає моделі проекту з усіма їх полями, які використовуються для більш простої взаємодії з СУБД PostgreSQL та являють собою кожен з таблиць у базі. Файл «search.py», за допомогою якого відбувається більша частина роботи з СУБД Elasticsearch. Файл «signals.py», де визначаються функції, викликання яких відбувається при певних умовах (наприклад, при зміні об'єкту під час його збереження або при настанні певної події). Файл «urls.py», у якому визначена вся адресація проекту, тобто перехід за кожним з посилань у строці адресу та «views.py», у якому виконується обробка усіх даних, що відправляються та отримуються з серверу. При цьому частину цього функціоналу виконуються в інших файлах, згрупованих за типом використання (наприклад, для роботи з тестами або коментуванню). Файл «auth_processor.py» відповідає за обробку реєстрації та авторизації користувачів, файл «comments_processor.py» відповідає за створення нових коментарів, файл «entity_processor.py» допомагає виконувати обробку даних для сторінок сутностей (тобто курсу, уроку, вебінару тощо), за допомогою файлу «profile_processor.py» відбувається уся взаємодія з профілем користувача, файл «search_processor.py» відповідає за роботу пошуку за різними сутностями на ресурсі, файл «tests_processor.py» допомагає в обробці та показі тестів після кожного з уроків, а також модулів після закінчення курсу. У файлах обробки сутностей та профілю також відбувається регулювання та налаштування кешу сторінок у СУБД Redis.

3.2 Структура класів ІС

Діаграма класів – статичне представлення структури моделі. Відображає статичні (декларативні) елементи, такі як: класи, типи даних, їх зміст та відношення.

На рис. 3.2 зображено діаграму класів системи, що показує організацію та взаємодію сутностей бази даних у СУБД PostgreSQL.

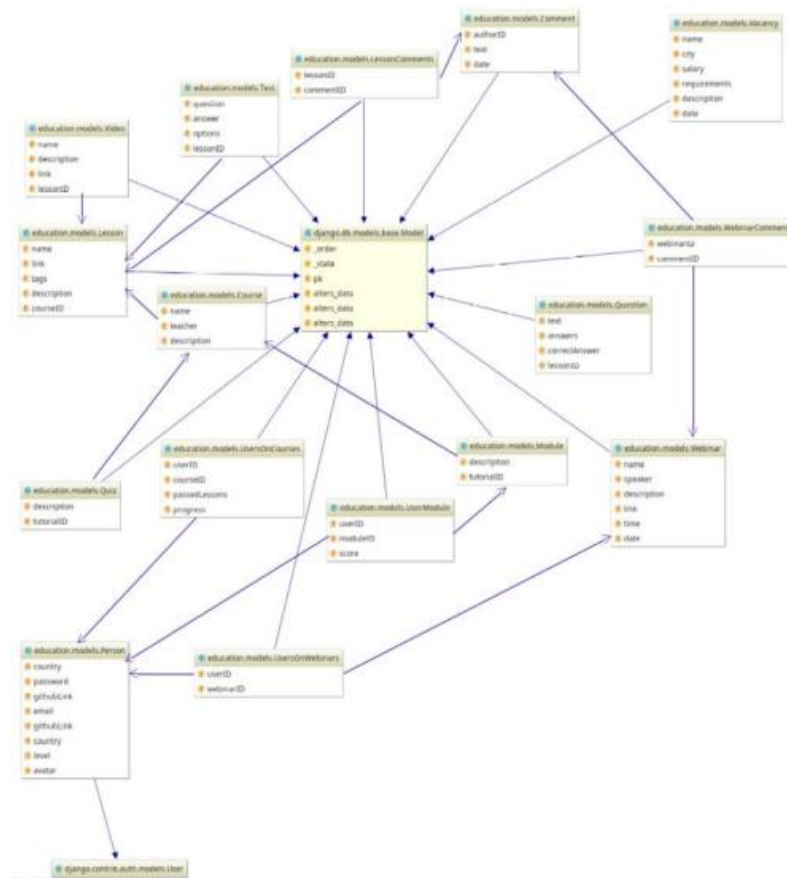


Рисунок 3.2 – Діаграма класів

Діаграма класів, також, може містити позначення для пакетів та може містити позначення для вкладених пакетів. Також, діаграма класів може містити позначення деяких елементів поведінки, однак їх динаміка розкривається в інших типах діаграм. Діаграма класів служить для

представлення статичної структури моделі системи в термінології класів об'єктно-орієнтованого програмування. На цій діаграмі показують класи, інтерфейси, об'єкти й кооперації, а також їхні відносини.

Усі моделі, окрім моделі користувача (Person), успадковуються від базового класу Django Models, а модель користувача успадковується від класу User, при цьому об'єкт класу Person не може бути адміністратором системи, на відміну від об'єкту класу User.

3.3 Інфраструктурне представлення ІС

Для забезпечення роботи інформаційної системи на серверній стороні необхідний комп'ютер з наступними технічними характеристиками як мінімум:

- 32-розрядний (x86) або 64-розрядний (x64) процесор з тактовою частотою 2 ГГц або вище;
- оперативна пам'ять не менше 3 Гб;
- вільне місце на жорсткому диску не менше 15 Гб;
- не менше 2 Гб для файлу підкачки.

Для розгортання необхідна операційна система Windows 7 або новіша, також можна використати ядро Linux версії 3.10 або вище. Для оперування даними використовуються PostgreSQL, Elasticsearch, Redis

ВИСНОВКИ

У кваліфікаційній роботі бакалавра було освітній веб ресурс для вивчення мов програмування. Запропонована інформаційна система стане у нагоді для осіб, які бажатимуть покращити свої знання сучасних інформаційних технологій, завдяки наявним на платформі курсам та завданням.

В роботі проведено аналіз аналогів розробленої інформаційної системи, були виділені переваги та недоліки для проектування та реалізації інформаційної системи. Були оглянуті та обрані технології для проектування та розробки інформаційної системи. Розроблена система вібрала у себе переваги аналогів та була покращена новими можливостями.

Після етапу огляду технологій та аналогів було спроектовано систему для вивчення мов програмування. Розроблені діаграми прецедентів, що надають повну інформацію зв'язку між користувачами та їх взаємодію із інформаційною системою. Спроектовані діаграми активності і послідовності роботи головних процесів програми, що надають інформацію про динамічне представлення інформаційної системи. Також були розроблені діаграми зв'язків у таблицях БД, спроектовані мінімальні необхідні вимоги до можливостей комп'ютеру для функціонування застосунку. А також було спроектовано алгоритми що описують різні сценарії використання освітнього ресурсу користувачами.

Було реалізовано та протестовано інформаційну систему. Реалізовані та описані усі компоненти та класи застосунку, створено діаграми головних класів для показу основних їх зв'язків та показу статичної моделі програми.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. UDACITY – Освітній ресурс. URL: <https://www.udacity.com/blog/2020/08/the-best-free-online-tech-co> (дата звернення 14.04.2023)
2. Coursera – Освітній ресурс. URL: <https://www.coursera.org/> (дата звернення 14.04.2023)
3. Khan Academy – Освітній ресурс. URL: <https://www.coursera.org/> (дата звернення 14.04.2023)
4. EdX – Освітній ресурс. URL: <https://www.edx.org> (дата звернення 14.04.2023)
5. Мова JavaScript та її можливості. URL: <https://sites.google.com/site/webtehnologiietawebdizajn/mova-javascript-ta-ieie-mozlivosti> (дата звернення 14.04.2023)
6. JavaScript. URL: <https://uk.wikipedia.org/wiki/JavaScript> (дата звернення 14.04.2023)
7. Нікітіна Т.С., Морозова О.І. Порівняльний аналіз продуктивності баз даних SQL та NOSQL. URL: <http://journals.nupp.edu.ua/sunz/article/download/1386/1179> (дата звернення 14.04.2023)
8. MongoDB Atlas. URL: <https://www.mongodb.com/> (дата звернення 14.04.2023)
9. Павловський В.І., Петрашенко А.В., Победа Д.В. Бази даних та засоби управління. Практикум. КПІ ім. Ігоря Сікорського. Київ : КПІ ім. Ігоря Сікорського, 2021. 112 с.
10. Підручник з Python. URL: <https://docs.python.org/uk/3/tutorial/index.html> (дата звернення 14.04.2023)
11. Django Project. Офіційний сайт. URL: <https://www.djangoproject.com> (дата звернення 14.04.2023)

12. Configure an interpreter using Docker. URL: <https://www.jetbrains.com/help/pycharm/using-docker-as-a-remote-interpreter.html> (дата звернення 14.04.2023)
13. Цеслів О.В. Основи програмування та веб-дизайн: Навч. посіб. – К.,2020. 149с. URL: https://ela.kpi.ua/bitstream/123456789/40499/1/OP_veb-dyzain.pdf (дата звернення 19.05.2023)
14. Шикуча О.М. Дизайн Web-сторінок:HTML+CSS: Навчальний посібник. К. ПДО, 2018. 60 с.
15. Django Documentation. URL: <https://docs.djangoproject.com/en/4.2/ref/databases/#postgresql-notes> (дата звернення 14.04.2023)
16. How to use Redis with Python. URL: <https://realpython.com/python-redis/> (дата звернення 14.04.2023)
17. Scrapy. Офіційний сайт. URL: <https://scrapy.org> (дата звернення 14.04.2023)
18. PostgreSQL. Офіційний сайт. URL: <https://postgresql.org> (дата звернення 14.04.2023)
19. Elasticsearch. URL: <https://pypi.org/project/elasticsearch/> (дата звернення 14.04.2023)
20. Redis. Вікіпедія. URL: <https://uk.wikipedia.org/wiki/Redis> (дата звернення 14.04.2023)

Д О Д А Т К И

ДОДАТОК А

МАКЕТИ ГРАФІЧНОГО ІНТЕРФЕЙСУ

1) Головна сторінка ресурсу:

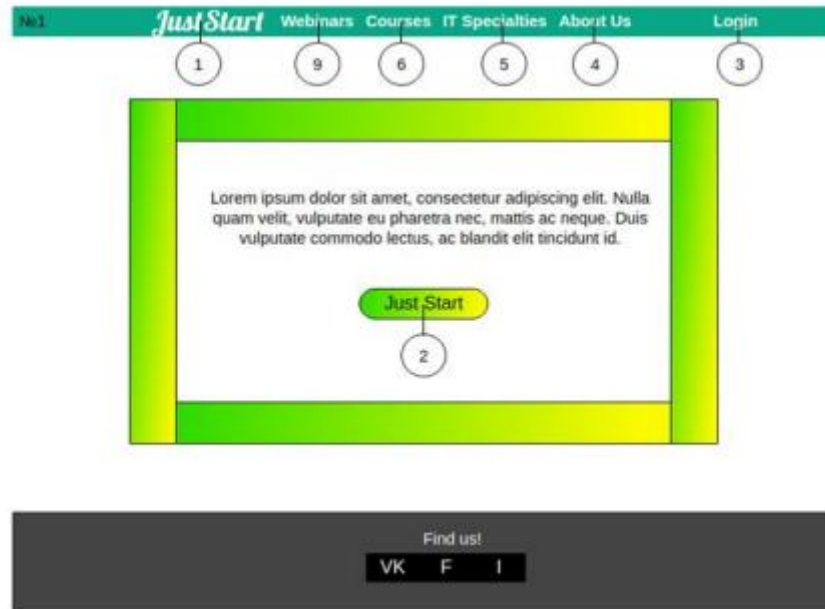


Рисунок А.1 – Головна сторінка

2) Сторінка реєстрації:

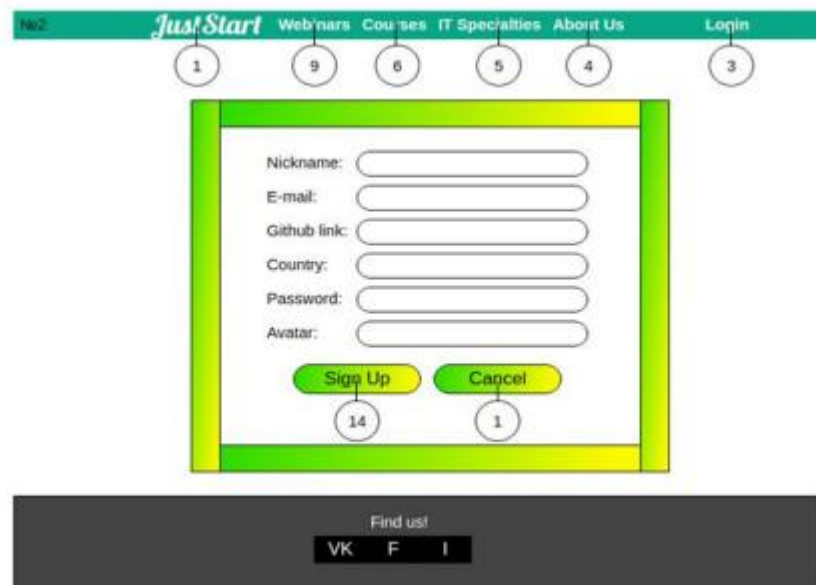


Рисунок А.2 – Сторінка реєстрації

3) Сторінка логіну:

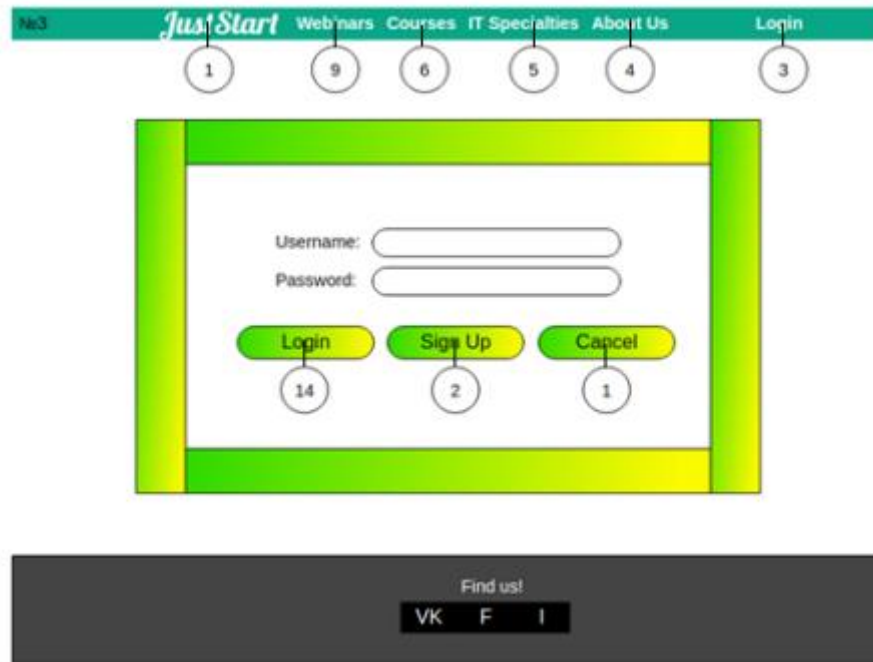


Рисунок А.3 – Сторінка логіну

4) Сторінка інформації про ресурс:

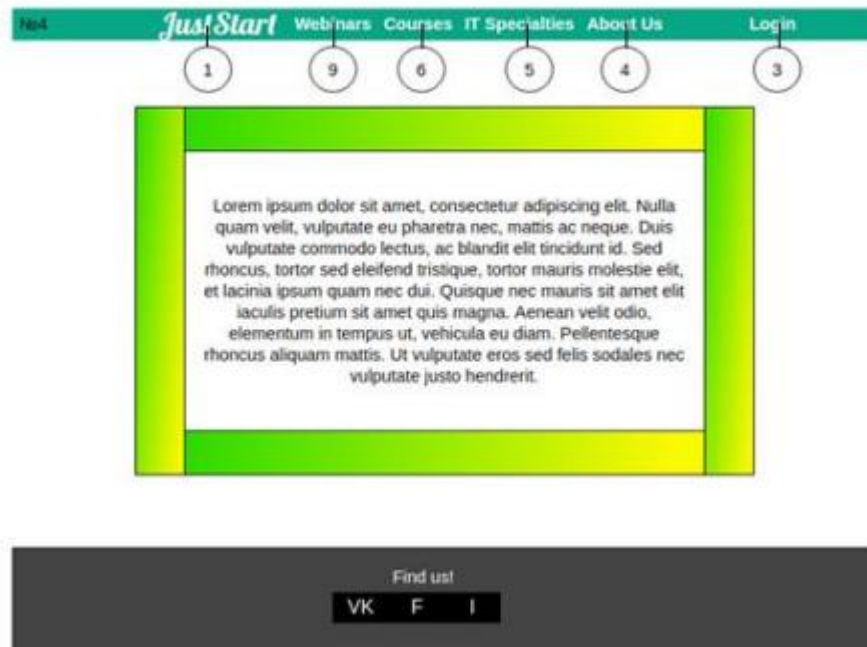


Рисунок А.4 – Сторінка інформації про ресурс

5) Сторінка зі списком вакансій:

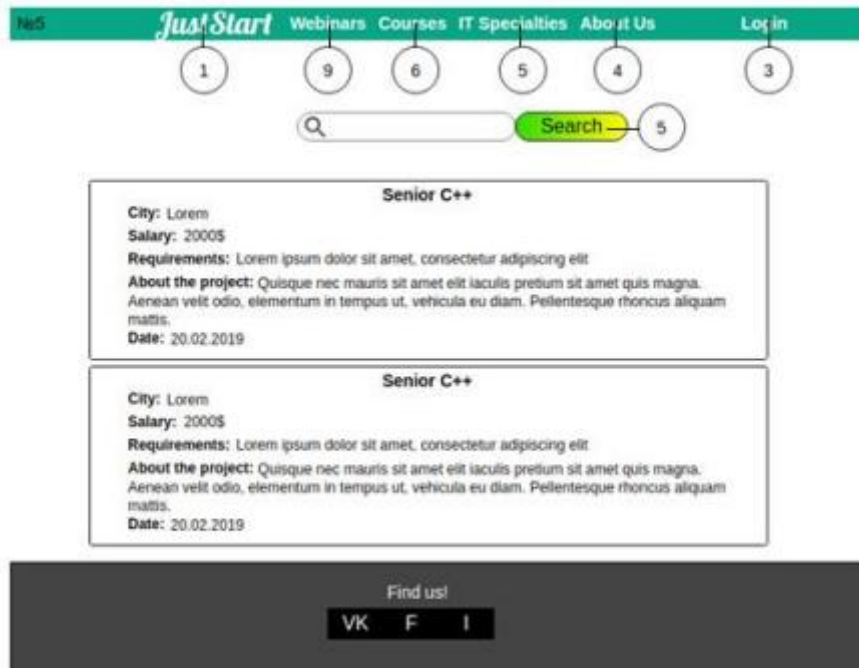


Рисунок А.5 – Сторінка зі списком вакансій

6) Сторінка списку курсів:

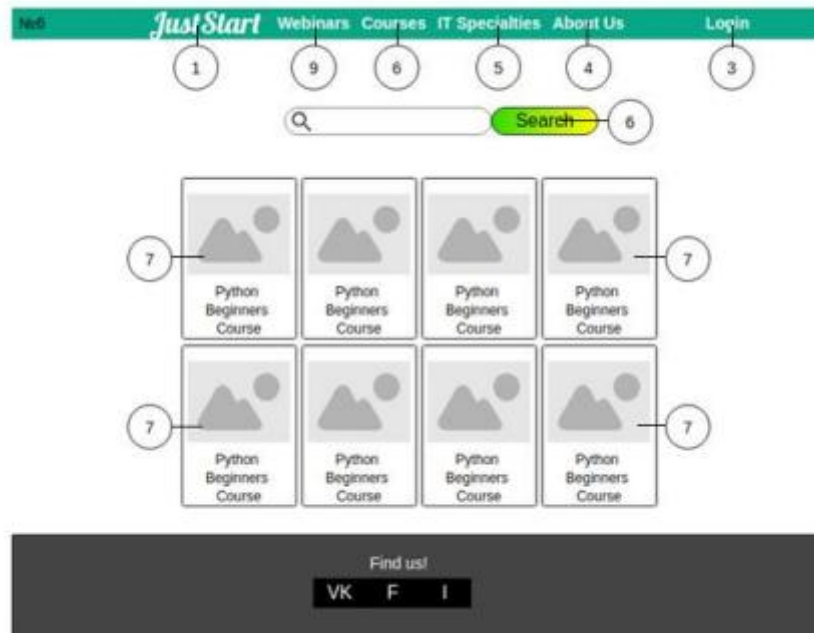


Рисунок А.6 – Список курсів

7) Сторінка вступу курсу:

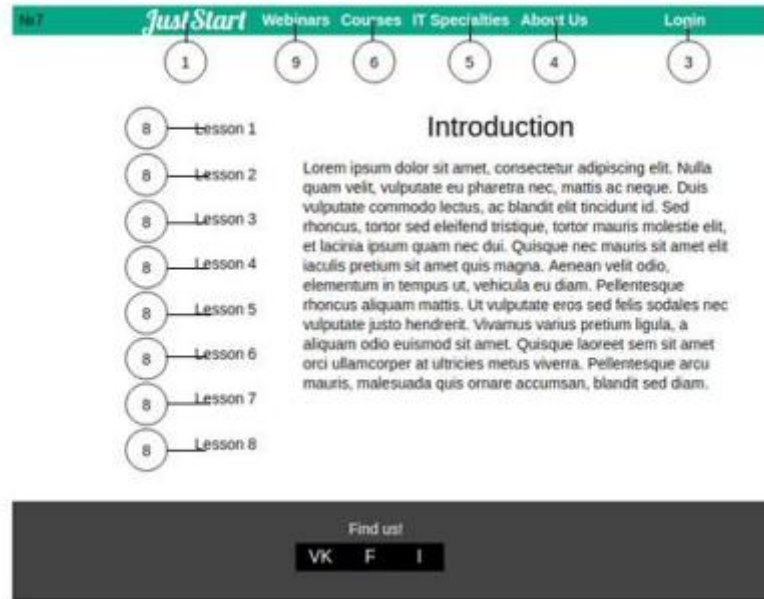


Рисунок А.7 – Початкова сторінка курсу

8) Сторінка уроку:

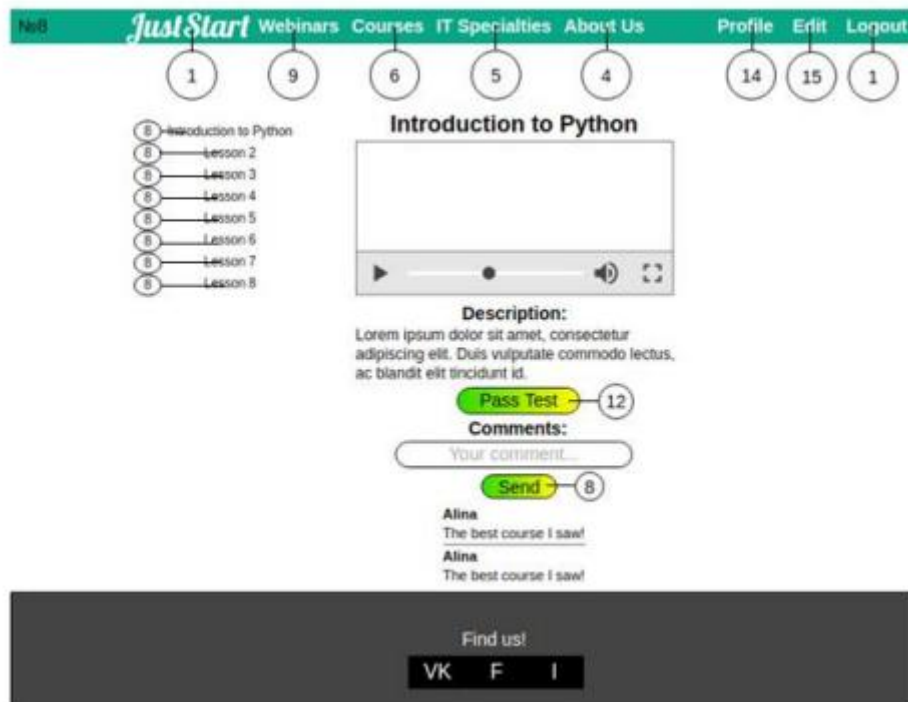


Рисунок А.8 – Сторінка уроку

9) Сторінка зі списком вебінарів:

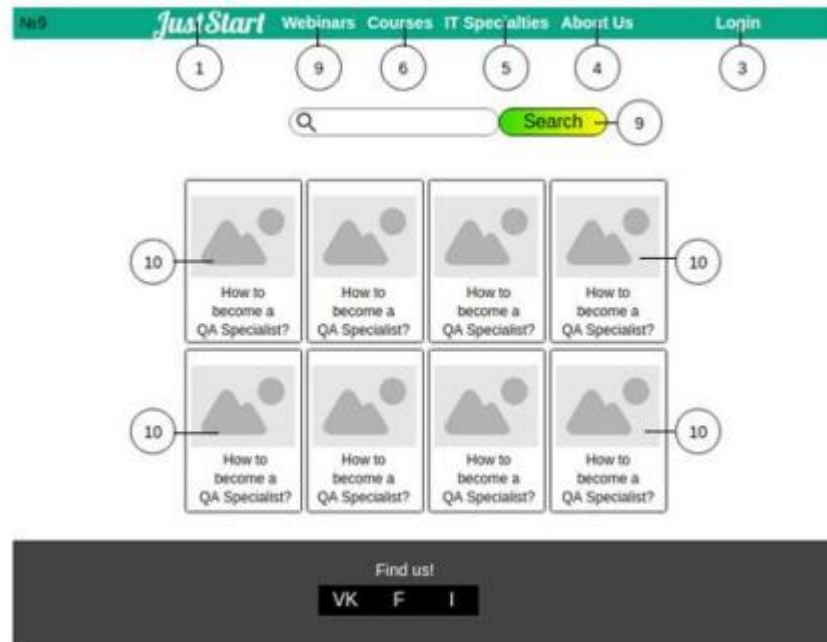


Рисунок А.9 – Сторінка зі списком вебінарів

10) Сторінка зі вступом вебінару:

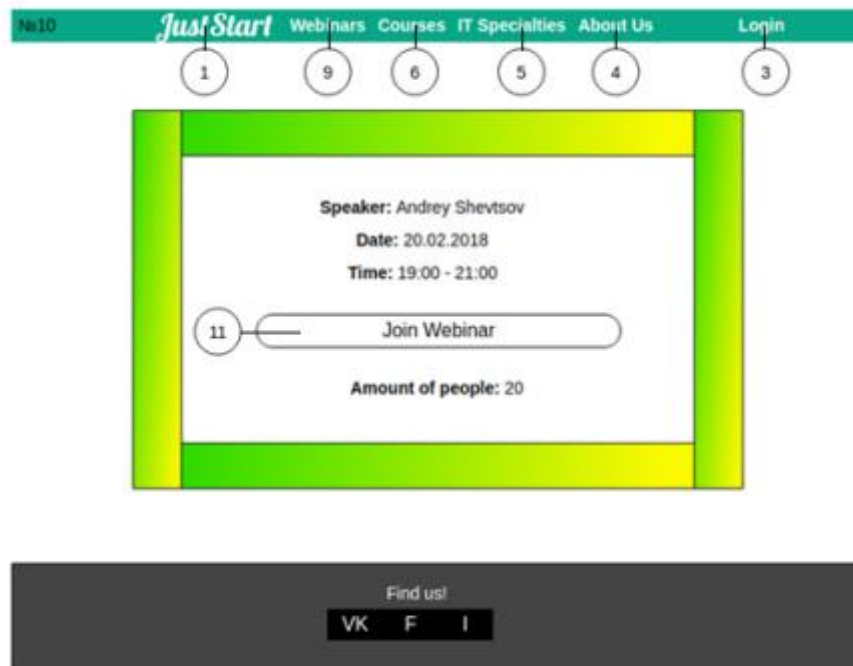


Рисунок А.10 – Сторінка зі вступом вебінару

11) Сторінка вебінару:

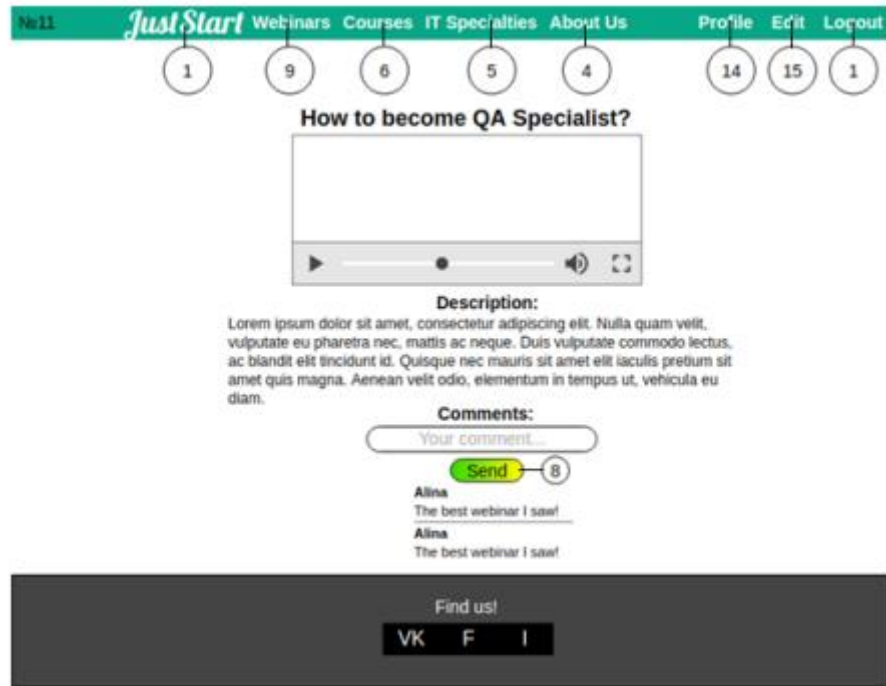


Рисунок А.11 – Сторінка вебінару

12) Сторінка з тестами:

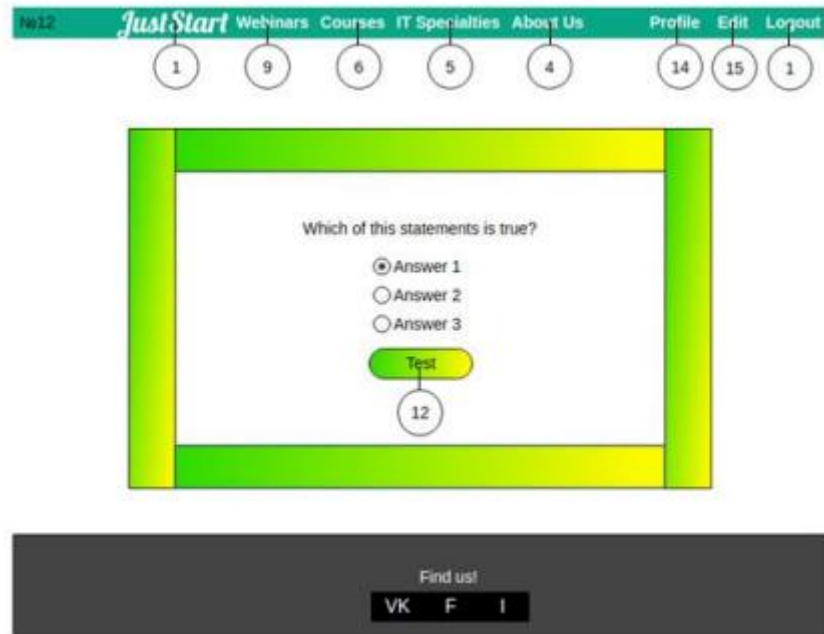


Рисунок А.12 – Сторінка тестування

13) Сторінка з результатом тестування:



Рисунок А.13 – Сторінка з результатом тестування

14) Сторінка профілю:

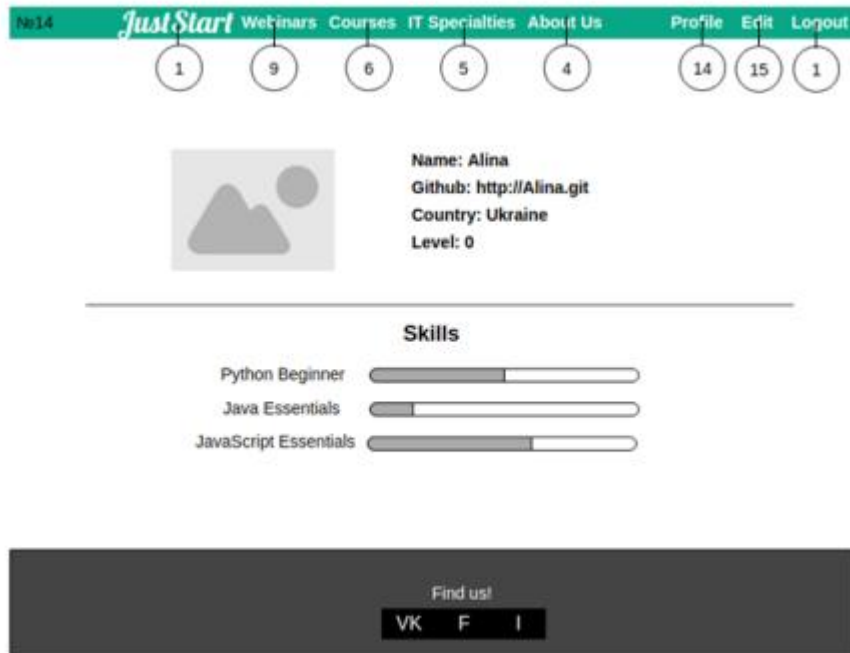


Рисунок А.14 – Профіль користувача

15) Сторінка редагування профілю:

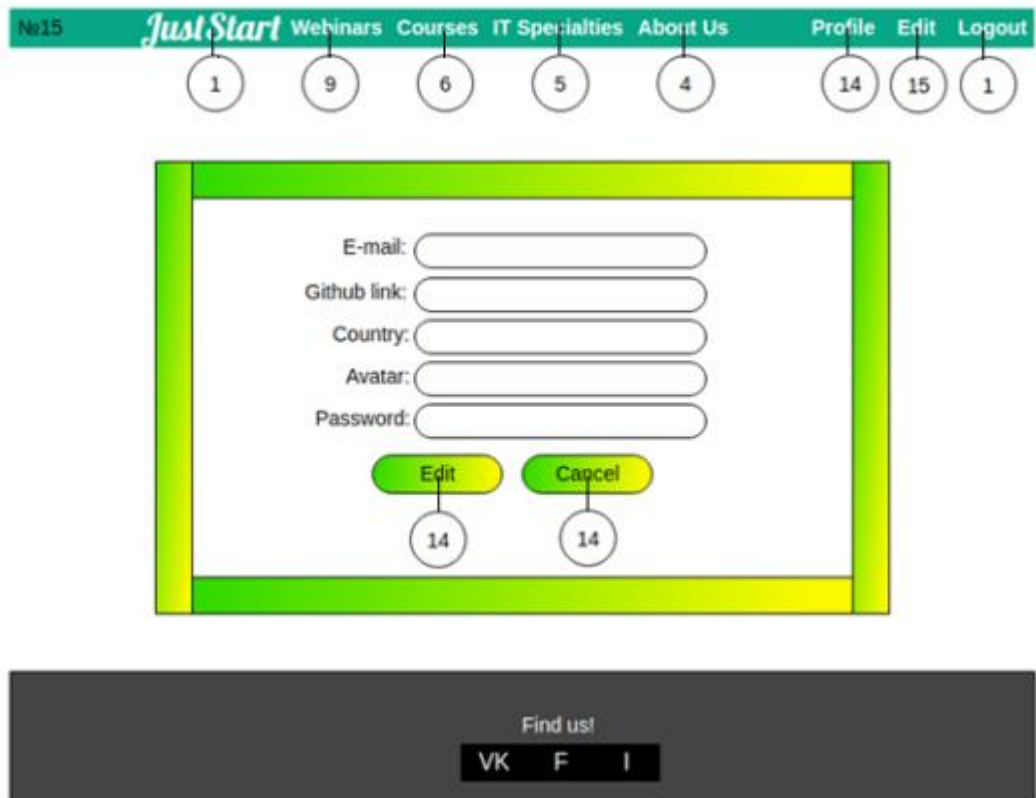


Рисунок А.15 – Сторінка редагування профілю

ДОДАТОК Б

ІНСТРУКЦІЯ КОРИСТУВАЧА

Реєстрація нового користувача (рис. Б.1).

1. У головному меню ресурсу натиснути кнопку "Just Start".
2. Заповнити усі поля, дотримуючись формату кожного поточного поля.
3. Натиснути на кнопку "Sign up".
4. Якщо усі поля заповнені коректно, користувач буде перенаправлений на сторінку показу профілю.
5. Якщо якісь поля заповнені некоректно, поряд з ними з'явиться повідомлення про те, що необхідно змінити для вдалого завершення реєстрації. Повторити пункт 3 після виконання змін даних.



Рисунок Б.1 – Сторінка реєстрації

Авторизація у ресурсі (рис. Б.2).

1. Натиснути на кнопку "Login" на панелі меню з правої сторони.
2. У формі, що відкрилася заповнити поля "Username" та "Password".
3. Натиснути на кнопку "Sign in".

4. Якщо з'явиться попередження, про те, що пароль не є вірним, необхідно ввести дані ще раз та виконати їх перевірку.

5. Якщо з'явиться попередження, про те, що такого користувача не існує, необхідно перевірити введені дані або натиснути на кнопку "Sign up" та зареєструвати нового користувача.

6. Якщо усі поля були заповнені вірно, користувач буде перенаправлений на сторінку показу профілю.



Рисунок Б.2 – Сторінка авторизації

Редагування профілю користувача (рис. Б.3).

1. Якщо на панелі меню з правої сторони є кнопка "Login", користувачу необхідно виконати авторизацію або реєстрацію. В іншому випадку необхідно натиснути на другу кнопку правої частини панелі меню.

2. У формі, що відкрилася, необхідно змінити лише ті поля, які потрібно, а усі інші залишити без змін. Поле паролю відображається пустим, але якщо ввести туди якісь дані, вони будуть збережені у якості паролю. При цьому необхідно дотримуватися формату даних у кожному з полів, щоб уникнути необхідності редагування введених даних.

3. Натиснути на кнопку "Save".

4. Якщо усі поля заповнені коректно, користувач буде перенаправлений на сторінку показу профілю.

5. Якщо якісь поля заповнені некоректно, поряд з ними з'явиться повідомлення про те, що необхідно змінити для вдалого завершення редагування профілю. Повторити пункт 3 після виконання редагування даних.



Рисунок Б.3 – Сторінка редагування профілю

Виконання пошуку за курсами та уроками.

1. Перейти на сторінку показу списку курсів.
2. У полі пошуку ввести необхідний запит.
3. Натиснути на кнопку "Search".
4. Якщо курси або уроки знайдені за описом чи назвою, то буде показаний список курсів, якому відповідають ці уроки.

5. Якщо нічого не було знайдено, користувачу буде видане повідомлення про неуспішний пошук.



Рисунок Б.4 – Сторінка пошуку за курсами та уроками