



Приватний заклад вищої освіти
Одеський технологічний університет «ШАГ»
Кафедра інформаційних технологій та фундаментальної підготовки

Випускна кваліфікаційна робота бакалавра

**«ІНФОРМАЦІЙНА СИСТЕМА УЗГОДЖЕННЯ ВЗАЄМОДІЇ
КОРИСТУВАЧІВ»**

на здобуття бакалаврського рівня вищої освіти
зі спеціальності «122 Комп'ютерні науки»

Виконавці проекту

№	П.І.Б.	Група
1	Сарієв Г.	КН-П-191
2	Кравчук Д.	КН-П-191
3	Лоскутников І.	КН-П-191
4	Денисов Д.	КН-А-191
5	Самбаева К.	КН-Д-191

Автор звіту

_____ Лоскутников Ілля Сергійович

АНОТАЦІЯ

Автореферат випускної кваліфікаційної роботи на здобуття бакалаврського рівня вищої освіти зі спеціальності «122 Комп'ютерні науки». - Одеса: ОТУШ, 2023 - 13 с.

Метою проекту є створення нової онлайн-системи для пошуку приватних фахівців, яка дозволить вирішувати побутові та бізнесові завдання. Платформа об'єднує замовників послуг, яким потрібно виконати роботу, та кваліфікованих фахівців, які шукають роботу або можливість заробітку. Дана система базується на існуючій платформі kabanchik.ua, проте функціонал та інтерфейс користувача були модернізовані. Платформа складається з декількох частин, кожна з яких має свою функціональність. Основні функції цих частин включають реєстрацію, автентифікацію, авторизацію, можливість надавати послуги від фахівців до замовників і навпаки, пошук, фільтрацію та сортування, систему відгуків та користувацьких уподобань.

Оцінка релевантності проекту базується на реалізації актуальних функцій, спрямованих на полегшення процесу пошуку роботи та підлеглих. Наша команда приділила особливу увагу різним категоріям людей з обмеженими можливостями зайнятості (наприклад, жінкам у декреті, особам з інвалідністю 3 групи, людям з неповною вищою освітою тощо). В результаті, ми створили більш деталізовану систему пошуку, що дозволяє здійснювати більш ефективне фільтрування та сортування. Це дозволяє користувачам отримувати саме ті результати, які відповідають їхнім потребам та вимогам.

Організація командної роботи в нашому проекті базувалась на Agile-методологіях. Цей підхід дозволив нам ефективно розподіляти завдання та забезпечувати продуктивну роботу в команді, а також вчасно виконувати поставлені завдання. Ми використовували зворотний зв'язок від нашого ментора, що дозволяв нам вчасно виявляти та виправляти помилки у нашій роботі. Для управління процесом розробки ми використовували програмний інструмент Jira. Також ми регулярно проводили онлайн-конференції один-два рази на тиждень для забезпечення зв'язку та спільної роботи команди.

ВСТУП_____5

Технічне завдання до проекту_____6

РОЗДІЛ 1._____11

РОЗДІЛ 2. Опис розробки_____14

РОЗДІЛ 3. Результат_____21

ВИСНОВКИ_____22

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ_____23

ВСТУП

Мета проекту: Створити сайт-клон "Кабанчика" під назвою "Busy Bee" для надання платформи, де користувачі зможуть розміщувати завдання та знаходити виконавців. Основна функціональність: Розміщення завдань, пошук виконавців, управління користувачами, платіжна система.

- Вимоги до бекенду:
- Архітектура:
- Використання монолітної архітектури.
- Технології:
- Мова програмування: C#.
- Фреймворк: ASP.NET WebApi.
- ORM: Entity Framework Core для роботи з базою даних PostgreSQL.
- Мапер: AutoMapper для перетворення даних між моделями.
- Валідація: FluentValidation для перевірки вхідних даних.
- Аутентифікація та авторизація: Використання Azure AD B2C для управління користувачами.
- Шаблон проектування: Clean architecture.
- Unit of Work / Repository Pattern: Для управління доступом до даних.
- Документація API: Використання Swagger для автоматичного генерування документації.
- Тестування: Використання Postman для тестування API.
- Вимоги до DevOps-інженера:
- Контейнеризація: Використання Docker для контейнеризації додатка.
- Хостинг: Розгортання на Azure Virtual Machine.
- Система контролю версій: Використання Azure Repos для управління кодовою базою.
- CI/CD інструменти: Використання Jenkins для автоматизації процесів CI/CD.
- Моніторинг: Налаштування Prometheus і Grafana для моніторингу системи.
- Проксі-сервер: Використання Nginx в якості проксі-сервера.
- Планування та розгортання інфраструктури проекту.
- Автоматизація процесів розгортання, моніторингу та масштабування системи.
- Встановлення та налаштування інструментів для CI/CD.
- Забезпечення безпеки та контролю якості проекту.
- Управління конфігурацією та забезпечення надійності системи.
- Вимоги до дизайнера:
- Проектування користувацького інтерфейсу та користувацького досвіду.
- Використання інструментів для дизайну: Фігма, Ілюстратор, Фотошоп.
- Вимоги до фронтенду:
- Фреймворк: Використання Angular для розробки фронтенд-частини сайту.
- Інтегроване середовище розробки: Використання WebStorm для розробки на Angular.
- Управління залежностями: Використання npm для управління пакетами.

Загальний опис проекту:

Опис функціональності:

Розміщення завдань:

- Створення нового завдання з вказанням категорії, опису та інших необхідних полів.
- Редагування та видалення завдання.

- Управління статусом завдання (відкрите, в роботі, завершене).

Пошук виконавців:

- Пошук виконавців за категоріями, навичками, локацією та іншими параметрами.
- Відображення списку доступних виконавців з основною інформацією та рейтингом.
- Фільтрація результатів пошуку та сортування за різними критеріями.

Оцінки та відгуки:

- Можливість оцінювати виконавців та залишати відгуки після завершення завдання.
- Відображення рейтингу та відгуків на профілях виконавців.

Цільова аудиторія:

- Визначення цільової аудиторії відповідно до виду завдань та послуг, пропонованих на сайті.
- Аналіз потреб і очікувань користувачів щодо функціональності та користувацького досвіду.

Розширюваність:

- Розробка модульної архітектури, що дозволяє легко додавати нові функції та можливості.
- Розгляд можливості інтеграції з іншими сервісами або платформами для розширення функціональності сайту.

Дизайн та користувацький інтерфейс:

Загальний стиль дизайну:

- Визначення бажаного стилю дизайну (мінімалістичний).
- Визначення колірної палітри, шрифтів та інших візуальних елементів, що відповідають обраному стилю.

Макети та прототипи:

- Створення макетів користувацького інтерфейсу для різних сторінок та функціональних елементів сайту.
- Розробка інтерактивних прототипів для демонстрації користувацької взаємодії та навігації по сайту.

База даних та зберігання даних:

Вибір та опис бази даних:

- Розробка структури бази даних, включаючи таблиці, зв'язки між ними та ключові поля.
- Врахування вимог до продуктивності та масштабованості бази даних.

Зберігання даних:

- Визначення даних, які повинні зберігатися в базі даних, включаючи інформацію про користувачів, завдання, коментарі тощо.
- Розробка моделей даних та визначення зв'язків між ними.

Безпека:

Аутентифікація та авторизація:

- Реалізація процесів аутентифікації користувачів з використанням Azure AD B2C.
- Встановлення прав доступу та ролей для різних типів користувачів.

Захист даних:

- Застосування заходів безпеки для захисту користувацьких даних, включаючи шифрування зберігання та передачі даних.

Система контролю версій:

Використання системи контролю версій:

- Інтеграція проекту з Git для управління версіями та історією змін.
- Створення репозиторію Git та налагодження процесу спільної роботи розробників.

Інфраструктура:

Контейнеризація:

- Використання Docker для контейнеризації бекенд- та фронтенд-частини додатка.
- Створення Docker-контейнерів для різних компонентів додатка та їх оркестрація.

Хостинг:

- Розгортання додатка на Azure Virtual Machine або іншому облачному сервісі.

CI/CD:

- Використання Jenkins для налаштування процесів CI/CD.
- Автоматизація процесів розгортання, тестування та випуску нових версій додатка.

Моніторинг:

- Налаштування Prometheus та Grafana для моніторингу системи.
- Відстеження метрик продуктивності, завантаження та інших параметрів для забезпечення стабільності та швидкодії сайту.

Управління проектом:

Створення плану проекту:

- Визначення етапів розробки, термінів та завдань.
- Встановлення пріоритетів та розподіл ресурсів.

Комунікація та співпраця:

- Встановлення засобів комунікації та співпраці між членами команди.
- Використання інструментів для спільного редагування та обміну документами.

Звітність:

Вихідний код повинен оформлятися у відповідності до загальноприйнятих стандартів, на зразок стандартів GNU <http://www.gnu.org/prep/standards/>, у тому числі з дотриманням вимог до коментування та документування коду.

Зберігання вихідних кодів ПЗ повинно здійснюватись за допомогою системи контролю версій (вибір конкретної системи узгоджується групою проекту). Структура файлів та каталогів повинна відбивати архітектуру ПЗ, на зразок:

- root
- Backend
- Web
- Підготовка звітів про стан проекту та результати роботи.
- Презентація результатів перед командою та замовником проекту.

РОЗДІЛ 1

Загальна характеристика системи:

Проект "Busy Bee" ставить за мету розробку інформаційної системи, яка надає платформу для узгодження взаємодії користувачів. Ця система дозволяє користувачам розміщувати завдання та знаходити виконавців. Основна функціональність включає розміщення завдань, пошук виконавців, управління користувачами та платіжну систему.

Для реалізації бекенду використовується монолітна архітектура, мовою програмування - C#, фреймворком - ASP.NET WebApi. Робота з базою даних PostgreSQL відбувається за допомогою Entity Framework Core. AutoMapper використовується для перетворення даних між моделями, а FluentValidation - для перевірки вхідних даних. Для аутентифікації та авторизації користувачів використовується Azure AD B2C. Проект організований за шаблоном Clean Architecture. Swagger використовується для автоматичної генерації документації API, а Postman - для тестування API. DevOps-інженер забезпечив контейнеризацію додатка з використанням Docker та його розгортання на Azure Virtual Machine. Управління кодовою базою здійснюється за допомогою Azure Repos, а процеси CI/CD автоматизовані з використанням Jenkins. Для моніторингу системи використовуються Prometheus і Grafana, а Nginx виступає в ролі проксі-сервера. Були проведені планування та розгортання інфраструктури проекту, а також забезпечено безпеку та контроль якості.

Дизайнер розробив користувацький інтерфейс та користувацький досвід за допомогою таких інструментів, як Figma, Illustrator та Photoshop.

Фронтенд розробляється з використанням фреймворка Angular, а розробка відбувається в інтегрованому середовищі розробки WebStorm. Управління залежностями здійснюється за допомогою npm.

Проект передбачає розміщення завдань з можливістю створення, редагування та видалення завдань, а також управлінням їх статусом. Крім того, користувачі можуть здійснювати пошук виконавців за різними параметрами, переглядати профілі виконавців з основною інформацією та рейтингом, а також залишати оцінки та відгуки.

Розробка системи проводиться модульно, що дозволяє легко додавати нові функції та розглядати можливості інтеграції з іншими сервісами або платформами для розширення функціональності сайту.

РОЗДІЛ 2

Процес розробки:

Початок розробки інформаційної системи для узгодження взаємодії включав дослідження функціональності та вимог до проекту. Кожен учасник команди приймав рішення щодо використання технологій та оптимізації процесу розробки, зокрема з метою більшої гнучкості для майбутніх змін.

На початку було визначено, що фронтенд буде розроблятися з використанням бібліотеки React, проте пізніше з'ясувалося, що фреймворк Angular краще відповідає потребам...

Далі було обрано назву проекту, і на її основі був визначений стиль усього проекту. Вся команда прийняла рішення залишитися на варіанті "Busy Bee", тому стиль проекту був вибраний відповідно до кольорів та концепції бджоли.

Спочатку було поставлено завдання розробити першу сторінку проекту. Бекенд відповідав за розробку системи зберігання та створення ієрархії "Категорії". Фронтенд займався створенням зручного інтерфейсу для відображення цих категорій з можливістю розширення кількості без потреби внесення змін у код проекту.

DevOps-інженер вибрав платформу для розміщення сервера та налаштував конфігурацію для автоматичного деплою проекту при оновленні як фронтенду, так і бекенду.

Після створення головної сторінки було вирішено додати основний функціонал - створення та прийняття замовлень на сайті. За допомогою шаблону, який дозволяв налаштовувати кожну категорію (сторінку створення замовлення), було вирішено додати функцію авторизації та реєстрації. Це було необхідно, оскільки потрібні були працівники та замовники, які могли взаємодіяти з новим функціоналом.

Після створення сторінок користувачів та налаштування авторизації та реєстрації, залишилося додати сторінки для виводу працівників, які бажали приймати замовлення певного виду робіт, а також сторінку замовлень від замовників.

РОЗДІЛ 3

Аналіз проекту:

Проект "BusyBee" спрямований на активну аудиторію, яка шукає швидку роботу, відповідну їх вмінням, або замовників, які потребують фахівців у конкретній галузі для швидкого виконання завдань.

При виборі типу проекту було вирішено, що існує велика кількість веб-сайтів, які дозволяють розміщувати вакансії на різні види робіт, але недостатньо ресурсів, що забезпечують швидке знаходження людей або зацікавлених осіб для конкретних завдань. У складні часи, коли багато людей не можуть знайти стабільну роботу, наш проект допомагатиме знайти швидку роботу та підтримку для тих, хто шукає можливостей.

З цією ідеєю був створений проект "BusyBee" - бажання допомогти всім, хто потребує допомоги, тим, хто бажає допомогти іншим, і тим, хто має почуття відповідальності за інших. Ми хочемо, щоб кожна людина мала можливість покращити своє життя, незалежно від його складності.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

Офіційна документація Angular: <https://angular.io/docs> - Це офіційний джерело документації Angular, де ви знайдете повну інформацію про фреймворк, його функціональність, практичні приклади та керівництва.

Angular University: <https://angular-university.io/> - Цей ресурс пропонує широкий спектр онлайн-курсів і матеріалів для вивчення Angular, включаючи розробку компонентів, роутинг, роботу з формами та багато іншого.

Angular in Depth: <https://blog.angularindepth.com/> - Це блог, який містить статті, підказки, практичні приклади та рішення проблем, пов'язаних з розробкою на Angular. Тут ви знайдете глибокий аналіз різних аспектів фреймворка.

Angular Material: <https://material.angular.io/> - Це офіційна бібліотека компонентів для Angular, яка дозволяє швидко створювати красивий та сучасний користувацький інтерфейс згідно з дизайном Material Design.

Stack Overflow: <https://stackoverflow.com/questions/tagged/angular> - Stack Overflow є популярним форумом для запитань та відповідей, де можна знайти відповіді на питання щодо розробки на Angular. Ви також можете задавати свої власні питання та отримувати допомогу від спільноти.

Офіційна документація Azure: <https://docs.microsoft.com/azure/> - Це офіційний ресурс документації Microsoft Azure, де ви знайдете повну інформацію про послуги та функціональність платформи, практичні приклади, керівництва та документацію API.

Stack Overflow: <https://stackoverflow.com/questions/tagged/azure> - Stack Overflow є популярним форумом для запитань та відповідей, де можна знайти відповіді на питання, пов'язані з Azure. Ви також можете задавати свої власні питання та отримувати допомогу від спільноти.