

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

Факультет комп'ютерних наук та кібернетики
Кафедра теорії та технології програмування

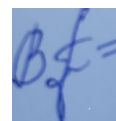
**Кваліфікаційна робота
на здобуття ступеня бакалавра
за спеціальністю 122 «Комп'ютерні науки»
на тему:
ВЕБСАЙТ ДЛЯ СТВОРЕННЯ РЕЗЮМЕ**

Виконав студент 4-го курсу бакалавра
Василь ГРИЦ



(підпис)

Науковий керівник:
доцент, кандидат фіз.-мат. наук
Віктор ВОЛОХОВ



(підпис)

Засвідчую, що в цій роботі немає
запозичень з праць інших авторів без
відповідних посилань.

Студент



(підпис)

Роботу розглянуто і допущено до
захисту на засіданні кафедри теорії та
технології програмування

« ____ » _____ 2023 р.,

протокол № _____

Завідувач кафедри

Микола НІКІТЧЕНКО

(підпис)

РЕФЕРАТ

Обсяг роботи: основний текст 30 сторінок, 49 ілюстрацій, 1 таблиця, 36 джерел посилань.

ВЕБСАЙТ, РЕЗЮМЕ, НАВЧАННЯ, ФОРМАТУВАННЯ ТЕКСТУ, КОМУНІКАЦІЯ, NODE.JS, JAVASCRIPT, EXPRESS.JS.

Об'єктом даної роботи є процес пошуку кандидатів чи вакансій, відповідно певних вимог. Предметом роботи є вебсайт для надання можливості детального опису інформації та доступного пошуку чи спілкування між кандидатами.

Метою кваліфікаційної роботи є проєктування та створення повноцінного вебсайту для навчання, опису інформації, пошуку та співпраці розробників.

Методи розроблення: проєктування баз даних, розробка серверної та клієнтської частини, та їхня взаємодія, використовуючи *Node.js*. Інструменти розроблення: інтегроване середовище розробки *Visual Studio Code*, *Postman*, *MongoDB*, *Vercel*, мова програмування *JavaScript*.

Результати роботи: розглянуто процес роботи вебсайту для створення резюме, проаналізовано наявні сервіси та їх недоліки для опису інформації та пошуку кандидатів, спроєктовано та реалізовано базу даних, розроблено інтерфейс користувача з можливістю спілкування, продуктивного навчання чи співпраці в певній сфері.

ЗМІСТ

СКРОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ.....	5
ВСТУП.....	6
РОЗДІЛ 1. ОГЛЯД НАЯВНИХ НА РИНКУ КОНКУРЕНТІВ.....	9
1.1 Вебсайт <i>Djinni</i>	9
1.2 Вебсайт <i>Glassdoor</i>	10
1.3 Вебсайт <i>Pracuj.pl</i>	10
1.4 Висновки.....	12
РОЗДІЛ 2. ОГЛЯД ВИКОРИСТАНИХ ТЕХНОЛОГІЙ.....	13
2.1 Технологія <i>Node.js</i>	13
2.2 Фреймворк <i>Express.js</i>	14
2.3 Документ – орієнтовна база даних <i>MongoDB</i>	16
2.4 Фреймворк <i>Bootstrap</i>	17
2.5 Вебплатформа <i>SendinBlue</i>	18
2.6 Вебплатформа <i>GitHub</i>	18
2.7 Сервіс <i>GoogleAuth</i>	19
2.8 Хмарні платформи <i>Vercel</i> та <i>Render</i>	19
2.9 Вебредактор <i>TinyMCE</i>	20
РОЗДІЛ 3. РЕАЛІЗАЦІЯ ТА ТЕХНІЧНІ ВИМОГИ ДО ВЕБСАЙТУ	21
3.1 Технічні вимоги до вебсайту	21
3.2 Реалізація серверної частини.....	22
3.2.1 Створення серверу і базової реєстрації з підключенням до <i>БД</i>	22
3.2.2 Робота з сесіями та основна розробка логіки вебсайту (<i>CRUD</i> операції).....	23
3.2.3 Розробка відправлення листів електронною поштою	24
3.2.4 Оптимізація вебсайту.....	25
3.3 Реалізація клієнтської частини	26
3.3.1 Базова розробка клієнтської частини	26
3.3.2 Основна розробка вебінтерфейсу	26
ВИСНОВКИ	30
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	31
ДОДАТКИ	34
Додаток А.1 Вигляд основної сторінки вебсайту для незареєстрованих користувачів	34
Додаток А.2 Сторінка реєстрації користувачів	37
Додаток А.3 Сторінка авторизації користувачів	37
Додаток А.4 Сторінка відновлення паролю	38
Додаток А.5 Приклад електронного листа після створення акаунту	38
Додаток Б Сторінка усіх резюме з додатковими можливостями на верхній панелі для зареєстрованих користувачів	39
Додаток В Сторінка окремого резюме	40
Додаток Г Сторінка улюблених резюме	41
Додаток Г.2 Приклад електронного листа після початку комунікації	43

Додаток Д.1 Сторінка створення резюме	44
Додаток Д.2 Сторінка створення резюме та демонстрація додаткових можливостей опису інформації	45
Додаток Е Сторінка усіх резюме, створених автором	46
Додаток Є Редагування вибраного резюме	47
Додаток Ж Сторінка профілю користувача	48
Додаток З Сторінка певної категорії.....	49

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ

- БД** – база даних;
- JS** – мова програмування JavaScript;
- IT** – інформаційні технології;
- CV** – curriculum vitae, стисле резюме;
- CRUD** – create - read - update - delete, операції створення - читання - оновлення та видалення;
- NPM** – Node Package Manager, менеджер пакунків для мови програмування JavaScript;
- HTTP** – HyperText Transfer Protocol, протокол передачі даних, що використовується в комп'ютерних мережах;
- API** – Application Programming Interface, прикладний програмний інтерфейс;
- JSON** – JavaScript Object Notation, текстовий формат обміну даними, заснований на JavaScript;
- URL** – Uniform Resource Locator, адреса будь-якого ресурсу у Всесвітньому павутинні: вебсторінки, сайту, документа, відео або файлу для завантаження;
- IDE** – Integrated Development Environment, інтегроване середовище розробки;
- SMS** – Short Message Service, коротке текстове повідомлення;
- Cookie** – невеликий файл даних з інформацією про користувача.
- PDF** – Portable Document Format, універсальний формат файлів, у якому зберігаються елементи вихідних документів, які створені в різноманітних програмах і на різних платформах.

ВСТУП

Оцінка сучасного стану об'єкта розробки. На сьогоднішній день в епоху цифрових технологій, кожен у своєму повсякденному житті використовує Інтернет та вебзастосунки для полегшення та пришвидшення повсякденних задач. Тому сфера розробки вебсайтів стає щораз то більш актуальною. Наявні на ринку вебсайти для створення резюме є малофункціональними, у більшості випадків вони дозволяють лише створювати *CV* у форматі *PDF*, без можливості форматування тексту. Спілкування з співробітником та надання додаткової інформації на цих вебсайтах, також викликають проблеми.

Актуальність роботи та підстави для її виконання. Створення вебсайту саме для опису більш детального резюме, з етапами навчання та іншим – є дуже актуальною для людей які шукають роботу не тільки в *IT*.

Варто зазначити, що саме при знайомстві з програмуванням часто складається ситуація, що початківці не мають конкретного плану, що саме і де вивчати. Важко знайти справді корисну інформацію, оскільки її багато, але й чимало застарілої та непотрібної. Початківці часто вивчають не те, що потрібно або ж витрачають багато часу на освоєння важкого матеріалу без ознайомлення базового.

Нижче наведено обґрунтування актуальності даного вебсайту:

- **Необхідність створення резюме:** за останні роки спостерігається зростання потреби створення електронного резюме, та особливо зараз, детального з описом досвіду, кроків навчання та іншої інформації.
- **Можливість структурованого навчання:** оглядаючи наявні резюме інших кандидатів, можна зберегти собі певні курси, покрокові етапи навчання чи ідеї проєктів, що дозволяє зрозуміти що саме і як вчити чи розробляти іншим працівникам.
- **Можливість пошуку кандидатів чи вакансій:** вебсайт також слугує одночасно місцем, передбаченим для пошуку кандидатів чи відповідних вакансій щодо певних вимог. Також, доступна можливість легкого спілкування між самими

кандидатами для подальшої співпраці, а не лише зв'язок кандидат-вакансія та навпаки.

Мета й завдання роботи. Метою кваліфікаційної роботи є розробка повноцінного вебсайту для створення детально описаного резюме, з реалізацією збереження, зміни та передачі інформації, а також з можливістю контактування з іншими клієнтами.

Для досягнення мети поставлено та виконано наступні завдання:

- Проаналізувати та познайомитися з процесом створення вебсайту, використовуючи лише *JS* фреймворки для серверної та клієнтської частини.
- Проаналізувати наявні вебсайти зі схожим функціоналом.
- Розробити вимоги до функціоналу системи.
- Підключити *БД* та створити моделі для кращого збереження даних та взаємодії з ними.
- Реалізувати авторизацію/автентифікацію, з перехопленням помилок, для майбутнього розділення резюме відносно чужих та своїй.
- Реалізувати базові *CRUD* операції для освоєння роботи з сервером.
- Реалізувати взаємодію клієнтської частини (*Bootstrap*, *HTML*, *CSS*) [1] з серверною частиною (використовуючи *Node.js* [2]), та роботою з *БД* (використовуючи *MongoDB* [3]).
- Реалізувати взаємодію користувачів та їх комунікацію для їх подальшого спілкування (використовуючи *SendinBlue* [4]).
- Додавання спеціального поля для користувача, для форматування тексту, додавання відео, посилання та ін. (*TinyMCE* [5]).
- Реалізувати авторизацію/автентифікацію використовуючи *GoogleAuth* [6].
- Розгорнути створений вебсайт на віддалений хостинг для загального користування, використовуючи хмарні технології *Vercel* [7] та *Render* [8].

Об'єкт, методи й засоби розроблення. Об'єктом даної роботи є процес розробки повноцінного вебсайту з розгортанням його на віддаленому хостингу.

Предметом роботи є система для перегляду та пошуку наявних резюме кандидатів або можливістю створити своє, та почати спілкування з вподобаними користувачами.

Розробка вебсайту складалася з кількох основних етапів:

1. Планування реалізації розробки;
2. Налаштування серверної частини з підтримкою роботи *БД*;
3. Реалізація логіки серверної частини, використовуючи *Node.js*;
4. Реалізація основної клієнтської частини для майбутніх сторінок та стилів.

Для створення вебсайту використовувались безкоштовні інтегровані середовища, а саме:

- Для основної розробки - (IDE) *Visual Studio Code* [9] з багатьма розширеннями для пришвидшення та покращення розробки;
- Для роботи з *БД* - платформа *MongoDB*;
- Для тестування серверної частини - платформа *Postman* [10];
- Збереження коду для загальної доступності - вебсервіс *GitHub* [11];
- Увесь код написаний на мові програмування *JS*, яка легко поєднує клієнтську та серверну частини, які, в свою чергу, написані на *JS* фреймворках.

РОЗДІЛ 1. ОГЛЯД НАЯВНИХ НА РИНКУ КОНКУРЕНТІВ

Ідея вебсайту для створення резюме є давно придуманою. Проте, більшість з вебсайтів, мають можливість просто робити CV у форматі PDF і генерувати посилання на це компактне резюме. Далі наведено кілька конкурентів та їх недоліки:

1.1 Вебсайт Djinni

На рис. 1 зображено вигляд сторінки вебсайту Djinni [12]. Даний вебсайт зараз являється одним з кращих в Україні для пошуку роботи та кандидатів в сфері ІТ. Проте, можна помітити, що тут інформацію можна прописати лише монолітним текстом та без додаткових відео чи таблиць.

djinni Відгуки Кандидати Вакансії Зарплати Не заповнений профіль ВГ Василь Гриц

Кандидати / Україна / .NET

Backend .NET Developer

Досвід роботи

Software engineer with almost four year of experience. I have experience of working with frameworks: .NET, .NET Core, ASP .NET (MVC/API), ASP .NET Core (MVC/API), Windows Forms, WPF, Entity Framework (6/Core), knowledge of different patterns such as Singleton, Repository, UnitOfWork, MVC e.t.c. I know how to work with Git repositories, I am a responsible and reliable person who is attentive to details. Hard-working and result-oriented so I am able to build a logical approach to achieve goals. Ready and willing to try me in different technologies and practices. Worked on different projects as a frontend & backend developer.

Навички

C# .NET Git Entity Framework OOP SQL LINQ REST API SOLID
 JavaScript Design Patterns HTML MVC Docker CSS microservices
 TypeScript jQuery WPF WinForms Elasticsearch Jenkins C #
 .net mvc Azure Source Tree

Мова спілкування
English

\$3500 / місяць

- Україна, Львів
- Можу працювати та бути на зв'язку
- 4 роки досвіду
- Англійська: Upper-Intermediate
- Переїзд в інше місто
- Віддалена робота
- Офіс

Опубліковано 17 травня 2023
 Середній час відповіді: 3 тижні
 Response rate 83%

Запропонувати вакансію

Рисунок 1 – Приклад вигляду вебсайту Djinni з сторони компанії

Також вебсайт дозволяє лише контактувати кандидатам з компаніями та навпаки, безпосередньо, щоб клієнти могли спілкуватись між собою, їм потрібно створити обліковий запис, як компанія, і запропонувати іншому співпрацю, що є не дуже зручним способом.

1.2 Вебсайт *Glassdoor*

Цей вебсайт слугує для пошуку роботи та оцінювання компаній, проте самим кандидатам, знову ж таки, важко розглянути резюме інших. На *рис. 2* наведено вигляд тексту для опису інформації, який є монотонним і важким для читання.

Вебсайт *Glassdoor* [13] є місцем для компаній, де вони можуть зарекомендувати свої вакансії та, очевидно, для кандидатів які шукають роботу. Проте, як і в попередньому варіанті, важко знайти резюме іншого кандидата, глянути його інформацію та комунікувати з ним.

The screenshot shows the Glassdoor website interface. At the top, there is a search bar with 'software developer' and 'Louisville, KY' entered. Below the search bar are navigation tabs for 'Jobs', 'Companies', 'Salaries', and 'Careers'. A filter bar shows 'All Job Types', 'Posted Any Time', '\$59K-\$120K', '25 Miles', and 'More'. The main content area displays a list of job listings under the heading '128 software developer Jobs in Louisville, KY'. The first listing is for 'Software Engineer - Remote' at 'iboss 2.9★' in Louisville, KY, with a salary range of '\$65K - \$102K (Glassdoor est.)' and a '24h' posting time. To the right of this listing is a detailed view of the job, including a 'Company Overview' for iboss, which describes its cloud security services and global presence. The overview text reads: 'iboss is a cloud security company that enables the modern workforce to connect securely and directly to all applications from wherever they work. Built on a containerized cloud architecture, iboss delivers security capabilities such as SWG, malware defense, RBI, CASB and data loss prevention to all connections via the cloud, instantaneously and at scale. This eliminates the need for traditional network security appliances, such as VPNs, firewalls and web gateway proxies, which are ineffective at protecting a cloud-first and mobile world. Leveraging a purpose-built cloud architecture backed by 230+ issued and pending patents and more than 100 points of presence globally, iboss processes over 150 billion transactions daily, blocking 4 billion threats per day. More than 4,000 global enterprises trust the iboss Cloud Platform to support their modern workforces, including a large number of Fortune 50 companies. To learn more, visit https://www.iboss.com/'.

Рисунок 2 – Приклад вакансії на вебсайті *Glassdoor*

1.3 Вебсайт *Pracuj.pl*

Цей вебсайт має свою гілку в Україні під назвою *РоботаUA* [14]. Він слугує одночасно і пошуком для вакансій, і має можливість створення резюме (див. рис. 3).

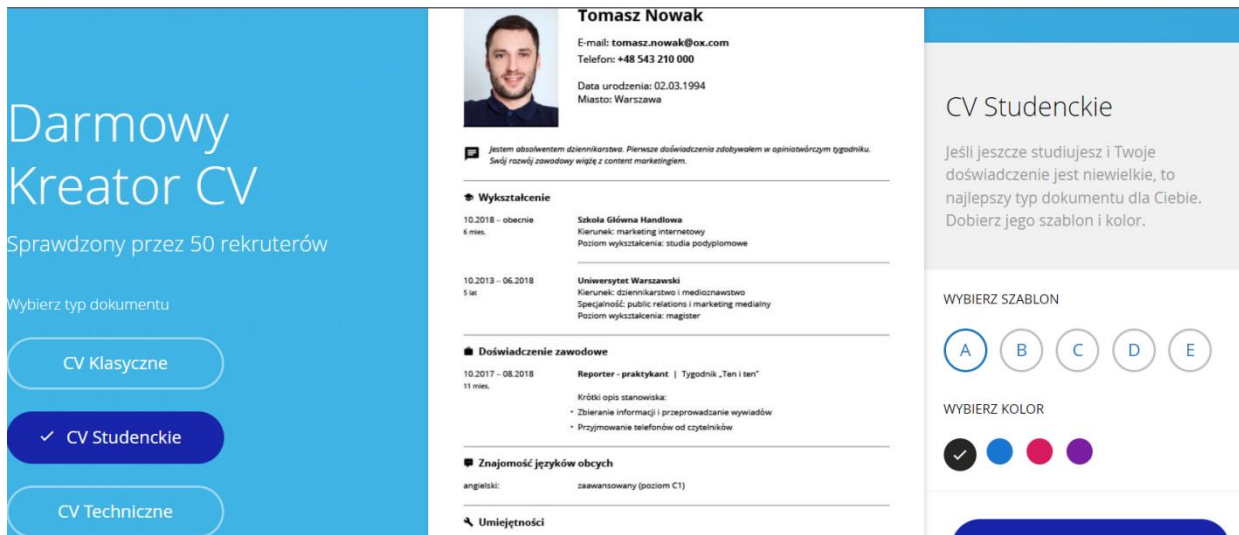


Рисунок 3 – Сторінка вибору вигляду CV на вебсайті *Pracuj.pl* [15]

Але можна помітити, що генерується CV у форматі PDF, де потрібно все стисло описувати, без удавання в деталі.

Проте, на українському вебсайті *РоботаUA* саме інформація для опису CV має можливість вибору форматування тексту, що дозволяє зробити опис більш придатним до читання і детально розповісти про самого кандидата (див. рис. 4).

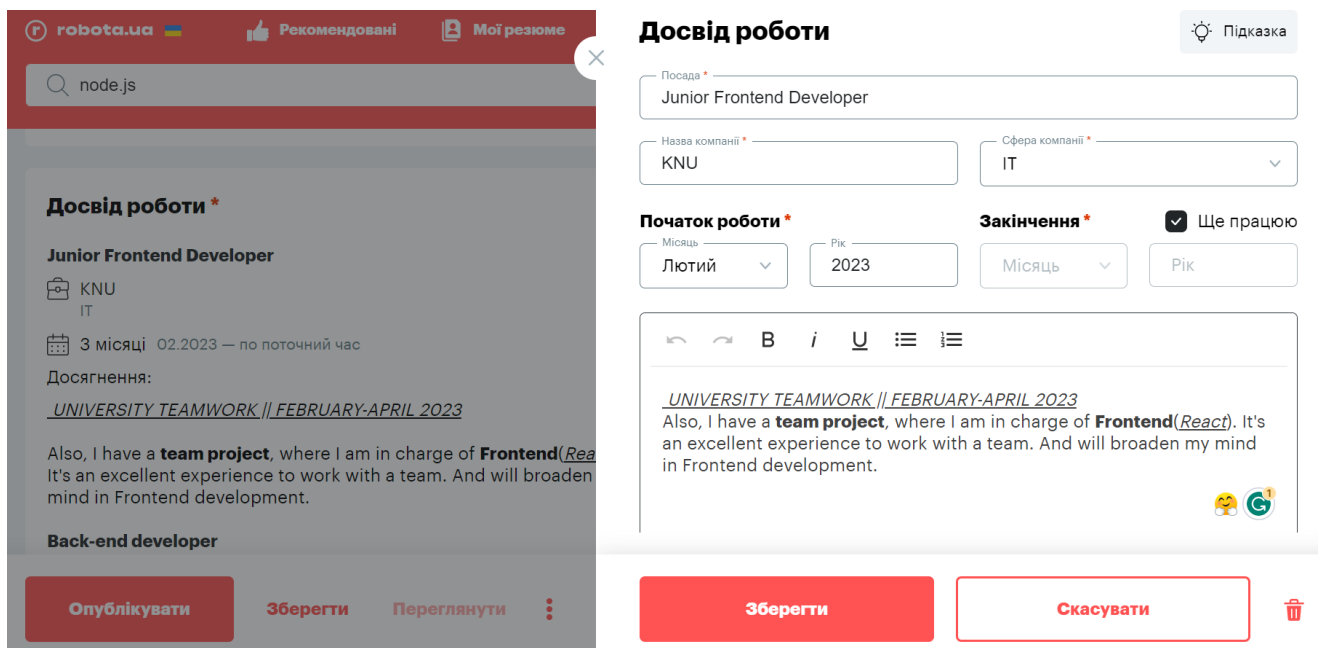


Рисунок 4 – Сторінка створення резюме на вебсайті *РоботаUA*

Можна помітити, що тут дозволено лише формувати текст без додавання відео, фото, таблиць та іншої потрібної інформації. Також, саме резюме є доступним лише для компаній, це означає, що розробники не можуть

переглядати резюме інших та бачити інформацію про навчання чи досвід, яка може бути корисною для співробітників чи початківців.

1.4 Висновки

Незважаючи на конкурентів, варто зазначити, що жодний вебсайт, який досліджено на просторах Інтернету, не має необхідного схожого функціоналу. Вебсайти стають щораз то більш корисними для пошуку роботи, навчання чи знаходження співробітників у певній галузі.

Також важливим недоліком вищезгаданих конкурентів є відсутність відсилання резюме за допомогою популярних соцмереж чи просто скопійованого посилання. Оскільки, часто користувачам необхідно поділитись інформацією про вакансію чи кандидатів, їм доводиться на цих вебсайтах виділяти весь маршрут для цього, що є не дуже зручним.

Дослідженні конкуренти також не мають можливості замороження резюме. У *Djinni* є, так званий, «пасивний пошук», який дозволяє компаніям бачити не нагальну потребу цього кандидата в пошуку вакансії. Тому виникає необхідність реалізації замороження резюме на певний період, який дозволяє показувати усім інформацію про себе, але не дозволено ділитись цим резюме з іншими та писати власнику. Це є дуже корисним пунктом для тих, хто розміщує своє резюме на вебсайті, але згодом не планує шукати співробітників чи вакансій, але бажає залишити свій шлях навчання для інших. Саме замороження резюме дозволяє не видаляти саме резюме, з можливістю в майбутньому його відновлення.

РОЗДІЛ 2. ОГЛЯД ВИКОРИСТАНИХ ТЕХНОЛОГІЙ

При розробці вебсайту використано наступні технології:

- технологія *Node.js*;
- фреймворк *Express.js* [16];
- документ – орієнтовна база даних *MongoDB*;
- фреймворк *Bootstrap*;
- вебплатформа *SendinBlue*;
- вебплатформа *GitHub*;
- сервіс *GoogleAuth*;
- хмарні платформи *Vercel* та *Render*;
- вебредактор *TinyMCE*.

2.1 Технологія *Node.js*

Node.js – це середовище виконання *JavaScript*, яке є крос-платформною з відкритим вихідним кодом, яке вперше випустив в 2009 році Раян Дал [17]. Це нова серверна платформа яка є побудованою на двигуні *Google Chrome JavaScript (V8 Engine)*, що дозволяє дуже швидко виконувати код. Мовою програмування є *JavaScript*, тому *Node.js* є фундаментальним в цьому плані, що він поєднує легко і швидко як і серверну, так і клієнтську частини і дозволяє їх розробляти, не змінюючи мову програмування.

Багатоплатформність. Багатоплатформна розробка в *Node.js* означає можливість створювання як зовнішніх, так і внутрішніх програм, не залежачи від програмного забезпечення операційної системи. Тобто це означає, що *Node.js* може працювати на *Windows*, *Linux* та *macOS* для створення масштабних та швидких мережеских програм. Також *Node.js* надає величезну бібліотеку різноманітних модулів *JS*, що спрощує роботу при розробці вебдодатків.

Відкритий вихідний код. Вихідний код *Node.js* є загальнодоступним, добре поясненим та оновлюється і підтримується розробниками з різних

куточків світу. Сам посібник і офіційна документація представлена різними мовами й українською в свою чергу [18].

Середовище виконання. *Node.js* є безпосередньо середовищем виконання *JS*. Тобто при написанні простого *JS* коду у якомусь текстовому редакторі, потрібне середовище для запуску коду. Для цього підходили вебпереглядачі як *Google* чи *Firefox*, тому до створення самого *Node.js*, сам код написаний на *JS* міг запускатись лише в браузері, що обмежувало його лише для розробки зовнішніх додатків. Тому саме створення *Node.js* допомогло вирішити проблему розробки внутрішніх додатків поза браузером.

***NPM*.** Дуже важливою частиною розробки на *Node.js* є використання *npm* [19], який є менеджером пакунків для полегшення, пришвидшення та покращення роботи програми. Це бібліотека та реєстр для програмних пакетів *JS*, яка має інструменти командного рядка, які, в свою чергу, допомагають встановити різні пакети та керувати їхніми залежностями. Майже на будь-які потреби є пакети з описаною документацією, яку оновлюють та покращують розробники ледь не щотижня. Також важливим аспектом є те, що можна створити свій пакет *npm*, який, на думку розробника, є чи унікальним, чи корисним, і використовувати його в своїх проєктах або ж розгортати для загального користування. *NPM* дозволяє *Node.js* легко взаємодіяти з БД, клієнтською, серверною частиною, відправкою електронних листів, тощо, використавши лише готовий пакет *npm*.

Отже, *Node.js* є однією з кращих технологій для написання вебдодатків, оскільки, швидко працює, має легко підтримувані фреймворки, і пов'язує всі частини розробки не змінюючи мови програмування.

2.2 Фреймворк *Express.js*

Фреймворк *Express.js* є дуже гнучким і водночас мінімалістичним для вебзастосунків, побудованих на *Node.js*. Він має безліч допоміжних методів для

роботи з *HTTP*, тому створити *API* можна легко, швидко та надійно. Сам *Express.js* є одним з найпопулярніших фреймворків для роботи з серверною частиною на *JavaScript*. Він легко співпрацює з клієнтською частиною та *БД*.

Оскільки *Express.js* має велику кількість можливостей розробки *HTTP* запитів та роботи з *API*, використано лише найпотрібніші для створення вебсайту, а саме:

Body-parser. Він використовувався як middleware *Express.js* для того, щоб зручно читати «тіло» вхідного об'єкта у форматі *JSON*.

Express-session. Оскільки, після кожного запиту *HTTP*, який є протоколом без збереження стану, що призводить до того, що в кінці самого запиту, сервер і клієнт забувають про один одного. Тому розроблено сесії, які мали містити певні дані про користувача і зберігали дозволи у пам'яті сервера або базі даних. Саме в цій роботі, сесії потрібні для фільтрування незареєстрованих та зареєстрованих користувачів, а також для обробки їхніх даних, відносно дій в цій сесії.

Passport.js. Це також middleware, але для автентифікації чи авторизації на *Node.js*, який допомагає легко та просто, взаємодіючи з *БД*, створювати і перевіряти користувача з вбудованими *Express.js* валідаціями.

Nodemon. Оскільки робота з вебзапитами *Express.js* вимагала часто оновлювати сервер, використано *nodemon*, який при збереженні будь-якого файлу коду автоматично це робив, економлячи багато часу.

Також важливо перелічити інші можливості, які пропонує *npm* для розробки вебсайтів на *express.js*, які були використані при розробці вебсайту, а саме:

Bcryptjs. Дуже хорошою практикою для роботи з вебсайтами є хешування та приховання певних даних користувача, до прикладу, коли треба зберегти їх в *БД*. Тому для цього використано готову *npm* бібліотеку *bcryptjs*, яка шифрувала реальні дані користувача, для подальшого збереження. Ця бібліотека має рівні складності шифру та звичайно методи розшифрування, наприклад, коли нам потрібно перевірити чи користувач ввів пароль правильно, а в *БД* у нас шифр, то

бібліотека має вбудовані методи перевірки, що є зручним, лаконічним і більш захищеним методом.

Dotenv. Також, оскільки, вебсайт включає в себе реєстрацію через *Google*, відправлення *email*, підключення до *БД* та секретні дані для шифру тощо, потрібно сховати файл з цими змінними від загального доступу. Тому використано пакет *dotenv*, який при створенні локального відповідного файлу, продовжує працювати з прихованими даними.

2.3 Документ – орієнтовна база даних *MongoDB*

Через просту роботу з колекціями та документами, аніж таблицями - можна без проблем взаємодіяти та витягувати дані з *MongoDB*, на інтуїтивному рівні. Сам вебсайт *БД*, є добре оснащеним для різних потреб та повністю замінює локальну версію цієї *БД*, що є зручним для розробників, які віддають перевагу *БД*, розміщеній в Інтернеті.

Для розробки вебсайту, створено безкоштовний кластер та підключено до самого коду, для подальшої взаємодії з *БД*. Згодом, були створені відповідні моделі, які слугують місцем збереження даних. Приклад, як ці дані збережені в *MongoDB* зображено на рис. 5.

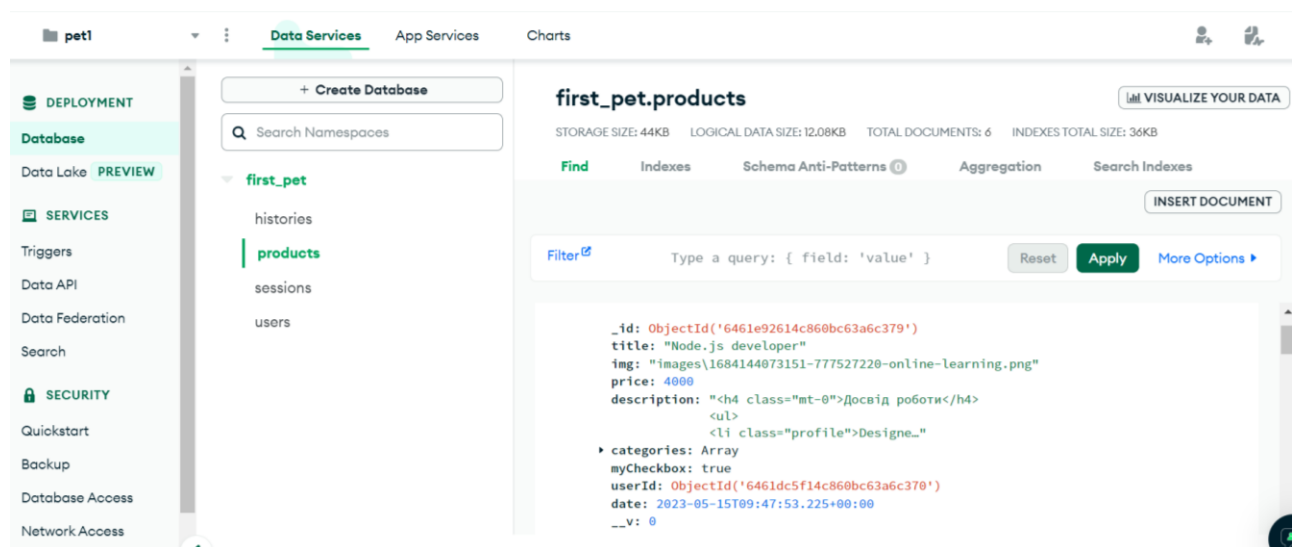


Рисунок 5 – Схематичне зображення моделі у *MongoDB*

Варто зазначити, що саме створення і підключення *БД* відбувається дуже швидко та просто, оскільки більшість розробників, які використовують *MongoDB*, розробляють продукти на *Node.js* та у IDE *Visual Studio Code*, тому створено максимально легку і зрозумілу взаємодію з цією *БД*.

2.4 Фреймворк *Bootstrap*

Bootstrap – це *JS* фреймворк для клієнтської частини з відкритим вихідним кодом, який слугує для створення швидких та адаптивних вебсайтів. *Bootstrap* використовують для клієнтської частини вебдодатків, які написані на різних мовах програмування, проте легка взаємодія з ними - робить його щораз то більше популярним для розробки продуктів. Проте, найкраще взаємодіє він з *JS*, тому при розробці, для клієнтської частини використано саме його.

Bootstrap включає в себе шаблони дизайну на основі *CSS* та *HTML* для різних форм, кнопок, навігацій, таблиць, зображень та багато інших – це полегшує і пришвидшує розробку вебдодатків. Також, при створенні адаптивних конструкцій, *Bootstrap* використовує плагіни *JS*, що є дуже корисними та зрозумілими можливостями при написанні клієнтської частини.

Варто зазначити, що для вебсайту, допоміжним пакетом для розмітки використано *EJS* [20]. Він прекрасно взаємодіє з необхідними *middlewares* серверу, та використовує *Bootstrap* – це дало вебсайту красивого візуального вигляду, та можливостей, не використовуючи серверної частини.

Також, *Bootstrap* розробники пропонують на їхньому сайті вже готові платні досконалі темплейти [21], які згодяться для клієнтів, які бажають лише розробляти серверну частину або ж просто, щоб пришвидшити розробку певних сторінок. Додатково, на сайті фреймворку можна знайти іконки [22] або ж за допомогою *Figma* [23], створити свої, які при використанні *Bootstrap*, чудово інтегруються, масштабуються та підключаються, на відміну від стандартних. Чудовою можливістю є також приклади вже готових сторінок, з базовими можливостями, проте найбільш популярними є при розробці вебсайтів [24].

2.5 Вебплатформа *SendinBlue*

Оскільки, вебсайт має необхідність відсилення електронних листів користувачу, при реєстрації, зміні паролю та ін., використано платформу для надсилення листів, а саме французьку платформу *SendinBlue* від компанії *Brevo* [25], створену в 2012 році, яка займається маркетингом, електронною комерцією та послугами повідомлень *SMS*, використовуючи електронні пошти.

Досліджено ще так само інші платформи, як *Nodemailer* [26], *Mailgun* [27] та прямий конкурент *SendGrid* [28]. Проте, у них набагато більше обмежень і складніший функціонал для безкоштовного використання платформи. Оскільки, вебсайт компанії *SendinBlue* має найбільш зрозумілу інструкцію для відсилення листів, саме на *Node.js* – використано саме його згодом, після порівнювання з іншими трьома платформами.

Також, саме підключення відбувалось завдяки пакету *SendinBlue (sib-api-v3-sdk)*, для якого спочатку створювався обліковий запис на вебсайті платформи і зберігався ключ *API* для подальшої взаємодії. Самі листи були розроблені простою *HTML* розміткою з усім необхідним наповненням. Користувач отримував лист при реєстрації облікового запису, при відновленні забутого паролю з необхідним підтвердженням та, коли інші розробники бажають поспілкуватися, після зацікавленості у резюме користувача.

2.6 Вебплатформа *GitHub*

GitHub є некомерційною компанією, яка пропонує хмарне сховище коду для розробників. Саме *GitHub* є однією з найбільших та найпопулярніших вебплатформ для спільної розробки програмного продукту.

При виконанні розробки вебсайту розділено роботу на окремі завдання. Згодом, створено кілька гілок репозиторію на потреби конкретних задач, що давало уявлення спільного виконання роботи. Лише при повному тестовому покритті, додано певний виконаний пункт до основної гілки. Та увесь код програми розгорнуто для загального доступу в *GitHub*.

2.7 Сервіс *GoogleAuth*

Важливим пунктом розробленого вебсайту була авторизація та автентифікація, використовуючи технологію *OAuth2* [29], а саме *GoogleAuth*.

Саме вхід через *Google* є найпопулярнішим і найшвидшим серед користувачів в браузері. Тому розроблений вебсайт мав реалізацію саме цього прикладу. Для реалізації цієї можливості ознайомлено з роботою з *Google Cloud*[30], та саме підключення до *JS*, використовуючи *Express.js*.

Варто зазначити, що при реєстрації нового користувача, за допомогою *GoogleAuth*, значення паролю за замовчуванням записувалось як *NULL*, що згодом призводило до помилки при шифруванні паролю. Тому цю проблему вирішено, додавши випадкову генерацію паролю, з вибраною складністю зламу, яке в майбутньому оброблене для входу у вебсайт або використане для зміни паролю.

2.8 Хмарні платформи *Vercel* та *Render*

Для того щоб розгорнути вебсайт для загального користування без прив'язки до локального комп'ютера використовуються хмарні платформи. Досить популярною платформою була *Heroku* [31], проте, нещодавно, розгортання проєктів стало лише платною можливістю. Тому вирішено шукати аналоги.

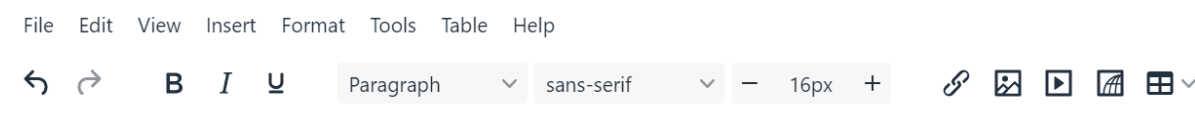
Render – це уніфікована хмара для створення та запуску віддалених додатків і вебсайтів з безкоштовними сертифікатами і захистом. Сам код проєкту підтягується, до прикладу, з *GitHub*. Перевагою цієї платформи є те, що кожного разу при запуску самого проєкту всередині платформи є демонстрація самих помилок чи інших потрібних повідомлень.

Vercel – американська хмарна платформа яка розробила технологію *Next.js* [32], яка має схожу ідею і функціонал як *Node.js*, проте поки є менш популярною та функціональною.

Також, код розгортається, підключивши *GitHub*, і перевагою тут, з попереднім варіантом, є те, що платформа пропонує не одне посилання, а декілька, для більш зручного користування. Порівнюючи з *Render* зрозуміло, що тут немає поля з виводом прямо у платформі, тобто при помилках, не буде зрозуміло що саме не так в самому коді. Проте, з іншого боку це дозволяє швидше запускати вебсайт, якщо немає нещодавніх оновлень у *GitHub*. Обидві платформи мають свої недоліки та переваги в певних моментах, проте є хорошими заміниками *Heroku*, і легкими в підключенні.


2.9 Вебредактор *TinyMCE*

Одним з ключових моментів розробки вебсайту є додавання можливості для користувача, більш детально і красиво описати свій шлях навчання, досвід з фотографіями, посиланнями, відео та ін. (див. рис. 4). Для цього завдання послугував вебредактор *TinyMCE*, який підтримується також *npm* і, за словами самих розробників платформи, є найбільш гнучким, масштабованим та настроювальним редактором у світі.



Lightweight. Loaded with features. Optimized for mobile.

Use TinyMCE as:

- A **basic** editor
- An advanced  editor
- A @collaborative editor
- A totally **<customized>** editor

Try out this demo to see how it works!

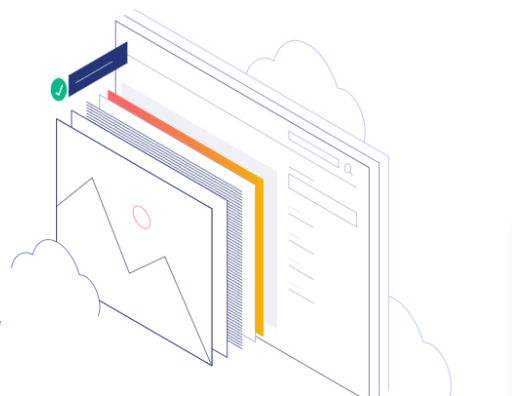


Рисунок 6 - приклад написання відформатованого тесту в *TinyMCE*

РОЗДІЛ 3. РЕАЛІЗАЦІЯ ТА ТЕХНІЧНІ ВИМОГИ ДО ВЕБСАЙТУ

3.1 Технічні вимоги до вебсайту

На сьогоднішній день, все більше людей в Інтернет, які використовують вебсайти для створення резюме. Проте, тут впливає кілька проблем, які дещо сповільнюють пошук необхідної інформації або ж взагалі не знаходження її.

Вебсайт повинен задовольняти наступні технічні вимоги для незареєстрованих користувачів:

- Інтерфейс основної сторінки: на цій сторінці має бути інформація про вебсайт, для чого він, що вміє і якими технологіями він розроблений. Також, важливо додати пункт про розробника, з можливістю поставити йому запитання або запропонувати ідею. Вебсайт повинен працювати і зображати наявні резюме і для незареєстрованих користувачів.
- Авторизація/автентифікація: вебсайт повинен мати можливість увійти чи зареєструватись стандартним способом або ж використовуючи вхід через *Google*. Для тих, хто забув пароль - має бути спосіб, щоб його відновити, через підтвердження електронною поштою. Також, після реєстрації, має приходити лист на пошту, що реєстрація пройшла успішно.
- Адаптивний дизайн з поясненнями: адаптивний дизайн необхідний для зручного користування вебсайтом на різних пристроях та розмірів екрану. А також потрібно додати на кожну сторінку пояснення як все працює і для чого потрібно.
- Обробка помилок та відображення повідомлень: вебсайт повинен мати механізм передбачення помилок та відображення порад чи повідомлень для користувача, що сприяє уникненню некоректної роботи вебсайту.
- Сортування категорій: важливим пунктом є сортування відповідних резюме по їхніх категоріях, та окреме сортування для власних резюме самого користувача, для швидкого доступу, видалення та редагування їх.

Технічні вимоги вебсайту для зареєстрованих користувачів:

- Створення власних резюме з усією необхідною інформацією.
- Збереження деяких резюме для майбутніх цілей, після їх перегляду.
- Контактвання з автором резюме, та, окремо, історія комунікації з можливістю чищення.
- Редагування власних резюме і можливість зробити резюме неактивним.
- Копіювання *URL* для кожного резюме в соцмережі, проте, якщо резюме неактивне, то цього не дозволяти.

3.2 Реалізація серверної частини

Для розробки основної логіки вебсайту використовувалось середовище *Visual Studio Code*, з багатьма допоміжними розширеннями та підключенням до *GitHub*. Це дозволяло розробляти програму різними гілками, в залежності від завдання, що допомагало уникати руйнування раніше написаного коду та імітувало роботу в команді. Вихідний код описано з кожним кроком роботи і додано в репозиторій *GitHub*[33].

Сама розробка була розділена на кілька етапів:

1. Створення серверу і базової реєстрації з підключенням до *БД*;
2. Робота з сесіями та основна розробка логіки вебсайту (*CRUD* операції);
3. Розробка відправлення листів електронною поштою;
4. Оптимізація вебсайту.

3.2.1 Створення серверу і базової реєстрації з підключенням до *БД*

Спочатку досліджено роботу *Express.js* саме для розробки серверу на *Node.js*. Після підключення необхідних залежностей запущено сервер на простому порті. Завдання вимагало розробити базову реєстрацію, тому для збереження користувачів, під'єднано проєкт до *MongoDB*, попередньо створивши необхідний кластер.

Реєстрація та автентифікація відбувалася на різних маршрутах, тобто сторінках. Для *БД* створено схеми моделей для майбутніх користувачів.

Для кожного введення даних додано валідацію та перехоплення помилок із зображенням відповідних повідомлень, для зручності користувача. Якщо ж при реєстрації дані введені коректно, перевірявся користувач в *БД*, чи він вже має обліковий запис, якщо ні - то додавався до *БД*, якщо так - виводилось повідомлення, що обліковий запис вже існує. Також, усі операції з сервером тестувались у *Postman*.

3.2.2 Робота з сесіями та основна розробка логіки вебсайту (CRUD операції)

Після підключення сесій, які слугували для збереження інформації, наприклад, який саме користувач взаємодіє з сервером в даний момент, розроблено можливість створення резюме, заповнивши кілька полів. Тут також додано валідацію для обов'язкових полів та їхніх типів.

Окремою сторінкою був вивід усіх створених резюме з необхідними для користувача полями. Згодом окремі кнопки, які відповідали за видалення резюме чи перенаправлення на сторінку редагування, знову ж таки взаємодіючи з *MongoDB*, та використовуючи *Express.js HTTP* - методи для цього, були додані на окрему сторінку, доступну лише автору резюме.

Згодом додано в моделі користувачів поля, які відповідали за *id* створених ними резюме, які їм сподобались. Тому розроблено сторінку улюблених резюме з можливістю видалення їх звідти. Використовуючи сесії, можна зрозуміти, який саме користувач створює резюме та перевіряти в *БД*, що він не додає в улюблені свої ж резюме. Також додано спеціальний *middleware*, який перевіряв, чи користувач знаходиться в сесії, якщо ні - йому не була доступна можливість створення, редагування та видалення резюме, а дозволено лише їхній перегляд.

Додано також більше можливостей, коли додається нове резюме: очікувана зарплата, назва, фото, і опис. Для редагування резюме усі введені раніше дані

були підтягнуті до поля редагування, щоб користувач заново все не писав, а просто змінив наявне. Також окремим пунктом є збереження фотографій. Сам шлях генерувався відповідно часу створення та випадкових значень. При видаленні резюме чи заміні фотографії – старе фото видалялось з БД, щоб не засмічувати пам'ять. Згодом створено окрему сторінку яка відповідає за профіль користувача та з можливістю редагування даних.

3.2.3 Розробка відправлення листів електронною поштою

Необхідність спілкування користувачів, призвело до розробки можливості надсилання листів електронною поштою. Для цього створено спеціальну електронну адресу, яка б могла відповідати лише за відправлення листів, від імені цього вебсайту. Як сказано вище, різні платформи тестувались для цієї потреби, проте найкращою і зручною платформою виявилась *SendinBlue*, яка прекрасно взаємодіє з *Node.js*.

Для відправлення листів настроєно саму платформу та базовий макет вигляду листів, з відповідними полями отримувача та теми листа. Оскільки користувач хотів би мати історію своїх комунікацій, додано відповідну сторінку, яка зображає кому, коли і на яке саме резюме надіслано листа. Сама можливість комунікації також розроблена таким чином, що лише резюме які додані до улюблених, можуть бути в майбутньому проінформовані в зацікавленості користувача. І, звичайно, додано можливість чищення історії комунікації, якщо це бажає користувач.

Важливим пунктом є також розробка методу відновлення паролю, яке потребувало підтвердження через електронну пошту, для захисту облікового запису користувача. Відправлення токена відбувалось за допомогою *jwt* методу *crypto* [34]. Яке генерувало відповідний токен, попередньо вибравши складність його та час існування, тобто якщо користувач не бажає оновлювати пароль, а токен прийшов на електронну пошту, то щонайменше через годину, натиснувши на посилання, відновлення не буде дійсним, бо токен буде не активним.

3.2.4 Оптимізація вебсайту

Додано окремо обробник не підтримуваного маршруту та сторінку для читання окремого резюме, де інформація більш детально описана. Також, при створенні облікового запису, автоматично додавалась аватарка, яку згодом можна змінити, теж саме зроблено для резюме, коли користувач не вибрав фотографію, вебсайт автоматично генерує її.

Одним з важливих моментів є додавання авторизації та автентифікації за допомогою *Google*. Така необхідність виникала для пришвидшення реєстрації облікового запису чи входження в нього. Окремо проведено дослідження як саме використовувати метод *OAuth*, а особливо для *Node.js* вебдодатків. Оскільки робота відбувається з сесіями, будь-яка зміна коду на серверній частині вимагала знову заходити в обліковий запис. Тут і є необхідний вхід через *Google*, який дозволяє не запам'ятовувати пароль та логін, а просто натиснути відповідну кнопку входу.

Оскільки, вебсайт може містити велику кількість резюме з різних професій, додано сортування по категоріях та окрема сторінка для зареєстрованих користувачів із їх власними резюме. Це дозволяло власнику бачити усі свої резюме в одному місці для швидкого редагування та видалення.

Також вебсайт розрахований для опису більш детального резюме, а отже, повинен мати можливість поширення цього резюме з іншими. Так як на більшості вебсайтів неможливо комусь поширити вакансію чи інформацію про автора, просто натиснувши кнопку (потрібно виділяти шлях власноруч, що є не зручним), прийнято рішення додати поширення *URL* резюме у соцмережі. Оскільки, більшість людей, зазвичай, спілкуються електронною поштою та месенджерами – додано до кожного резюме відповідні кнопки з соцмережами, які відразу при натисненні мають збережене посилання на резюме і чекають, куди саме користувач хоче відправити. Окремою кнопкою додано просто копіювання *URL* резюме, якщо потрібно поділитись посиланням в яесь інше місце. Така можливість поширення активна і незареєстрованим користувачам,

що дозволяє, наприклад, людям, які ще не встигли зареєструватися, переглядати резюме та інформацію про досвід інших, і ділитись таким же посиланням для своїх благ.

В кінці, готовий протестований вебсайт був розгорнутий на віддалених хостингах, попередньо приховавши ключі (бази даних, токенів, Google, та ін.) та налаштувавши сам код для майбутнього розгортання в хмарних технологіях.

3.3 Реалізація клієнтської частини

Клієнтська частина мала менше етапів розробки, проте завжди потребувала велику необхідність в покращенні функціональності.

3.3.1 Базова розробка клієнтської частини

Спочатку створено прості сторінки, без стилів та форматування для тестування самої серверної частини. Згодом, для демонстрації даних отриманих від серверу, використано стилі *EJS*. Ця технологія дозволяла робити все необхідне на базовому рівні. Також, додано демонстрацію вхідних даних та зображення повідомлень при їх некоректності, наприклад, при створення резюме.

Для демонстрації усіх та окремо кожного резюме, сплановано та розміщено залежності та інформація, яка дозволяла легко орієнтуватися та швидко знаходити потрібні деталі резюме. Потім, додано стилі для електронного листа, який відправлявся користувачу при реєстрації, відновленні паролю чи комунікації з іншими.

3.3.2 Основна розробка вебінтерфейсу

Згодом, вирішено підключити *Bootstrap*, для кращої демонстрації та зовнішнього вигляду, як результат, перероблено сторінки авторизації та автентифікації, додано інтерактивну верхню панель для більш зручного перемикання між сторінками.

Майже кожна сторінка, форма, кнопка та ін., набули красивішого вигляду після додавання стилів та кращого форматування. Також, додано основну привітальну сторінку, в якій навіть для незареєстрованих користувачів є пояснення що це за сайт, для чого він, як і чим розроблений, та окремо є стаття про автора, яка надає можливість зв'язатись з ним.

Для демонстрації усіх резюме, додано також спеціальні маркери для позначення резюме, які створив користувач. Також, для кожної категорії, по якій резюме сортуються, додано марки для кращого розуміння. На кожній сторінці є також пояснення та демонстрації, що саме тут відбувається. Згодом, додано час створення резюме та, для сторінки історії, демонстрація у вигляді таблиці, що дозволяло легко розуміти кому і що відправлено, і додаткова кнопка для чищення історії.

Також додано пристосовування вигляду вебсайту під різні екрани та пристрої.

Оскільки, вебсайт відправляє деяку інформацію власною електронною поштою – створено логотип, який відображався всюди, як на основній сторінці, так і в електронних листах та ін. Згодом, додано більш темну тему вебсайту і змінено усі виведення відносно стилю логотипу та базової сторінки. Також, додано кращий вигляд кнопок для поширення резюме у соцмережі з відповідними іконками.

Згодом розроблено те, що робить цей вебсайт більш кращим, ніж інші платформи для створення резюме та пошуку кандидатів: додано краще форматування тексту для опису детальної інформації, з використанням стилів, та різних додаткових елементів як таблиці, кольори, відео, та ін., використовуючи *TinyMCE*.

І, наприкінці, додано можливість зробити своє резюме неактивним, не видаляючи його. Це необхідно, коли автор бажає обмежити доступ до комунікації, проте хоче поділитися досвідом своєї розробки з іншими, оскільки вона може бути корисною. Неактивні резюме відображаються у такому сірвато-трьмяному фоні, з додатковим підписом замість кнопок поширення та додавання

до вибраних. Також, якщо резюме додане до когось у вибране, і воно згодом стало неактивним, то і в збереженому також буде неактивним, що означає, що поки зв'язатись з власником заборонено. Варто зазначити, що увесь інтерфейс та робота вебсайту винесена окремо в Додатки.

Нижче наведено кілька прикладів щодо покращення та оптимізації вебсайту (див. табл. 1).

Таблиця 1 – Опис ідей щодо покращення та оптимізації вебсайту

Коротка назва	Опис
<i>Створення облікового запису адміністратора</i>	Ідея полягає в тому, щоб адміністратор міг проводити перевірку створених резюме, якщо там використано цензуру, спам чи інші заборонені речі. Вже після цього, він би дозволяв додавати резюме для загального доступу
<i>Cookie та дозвіл на оброблення даних</i>	При реєструванні користувача, планується додати інформація на обробку даних та файл cookie, для майбутньої взаємодії
<i>Статистика з діаграмами</i>	Планується додавання на основну сторінку діаграм, для демонстрування популярності технологій, середньої очікуваної зарплатні, тощо
<i>Активність резюме</i>	Важливим пунктом в майбутньому є розробка активності резюме, тобто демонструвати, скільки користувачів (не залежно чи зареєстровані) переглянули певне резюме, та скільки почали спілкування з автором. Це дозволить краще розуміти популярність певних резюме і полегшить пошуки кандидата
<i>Пошук за додатковим сортуванням</i>	Також, важливо додати пошук, який би пробігався по назвах резюме та їх наповнення. Згодом можна

	<p>додати сортування по популярності та дати створення резюме</p>
<p><i>Стеження за користувачем</i></p>	<p>Однією з найкращих майбутніх ідей є відстеження користувача. Це дозволить завжди отримувати будь-які новини про оновлення, додавання нових резюме тощо, що зробить вебсайт більш функціональним</p>
<p><i>Чат</i></p>	<p>Оскільки, комунікація відбувається через електронні пошти, чудовим вирішенням є створення чату прямо у вебсайті, який буде показувати історію комунікації, більше інформації про самого автора, а також пришвидшить саме спілкування</p>
<p><i>Замороження облікового запису</i></p>	<p>Останньою ідеєю є замороження облікового запису. Тобто, якщо користувач бажає на певний момент призупинити отримування повідомлень, чи іншої інформації, без видалення самого облікового запису. Це буде потрібним тим, у кого багато резюме реалізовано на вебсайті, та кожне резюме робити неактивним буде недоречно</p>

ВИСНОВКИ

Під час розробки цього вебсайту, досліджено принцип роботи та створення вебсайтів використовуючи основні технології *Node.js*, *Express.js*, *MongoDB* та допоміжні. Головною метою роботи було створення функціонального вебсайту для зручного пошуку співробітників та детального опису при створенні резюме.

Для досягнення поставленої мети були реалізовані наступні завдання:

- Проведено дослідження наявних конкурентів та визначення їхніх недоліків.
- Розроблено структуру вебсайту з підтримкою *БД*.
- Реалізовано функціональності сортування резюме, форматування тексту, легкого копіювання *URL* резюме у соцмережі, контактування між клієнтами та можливістю деактивувати резюме.
- Забезпечено інтуїтивно зрозумілий користувацький інтерфейс, для швидкого і легкого користування вебсайтом.
- Розгорнуто вебсайт на віддаленому хостингу для загального користування.

Результатом роботи є зручний та багатофункціональний вебсайт, який буде корисним для багатьох користувачів та компаній, для опису детального резюме, навчання, співпраці, пошуку кандидатів чи вакансій.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

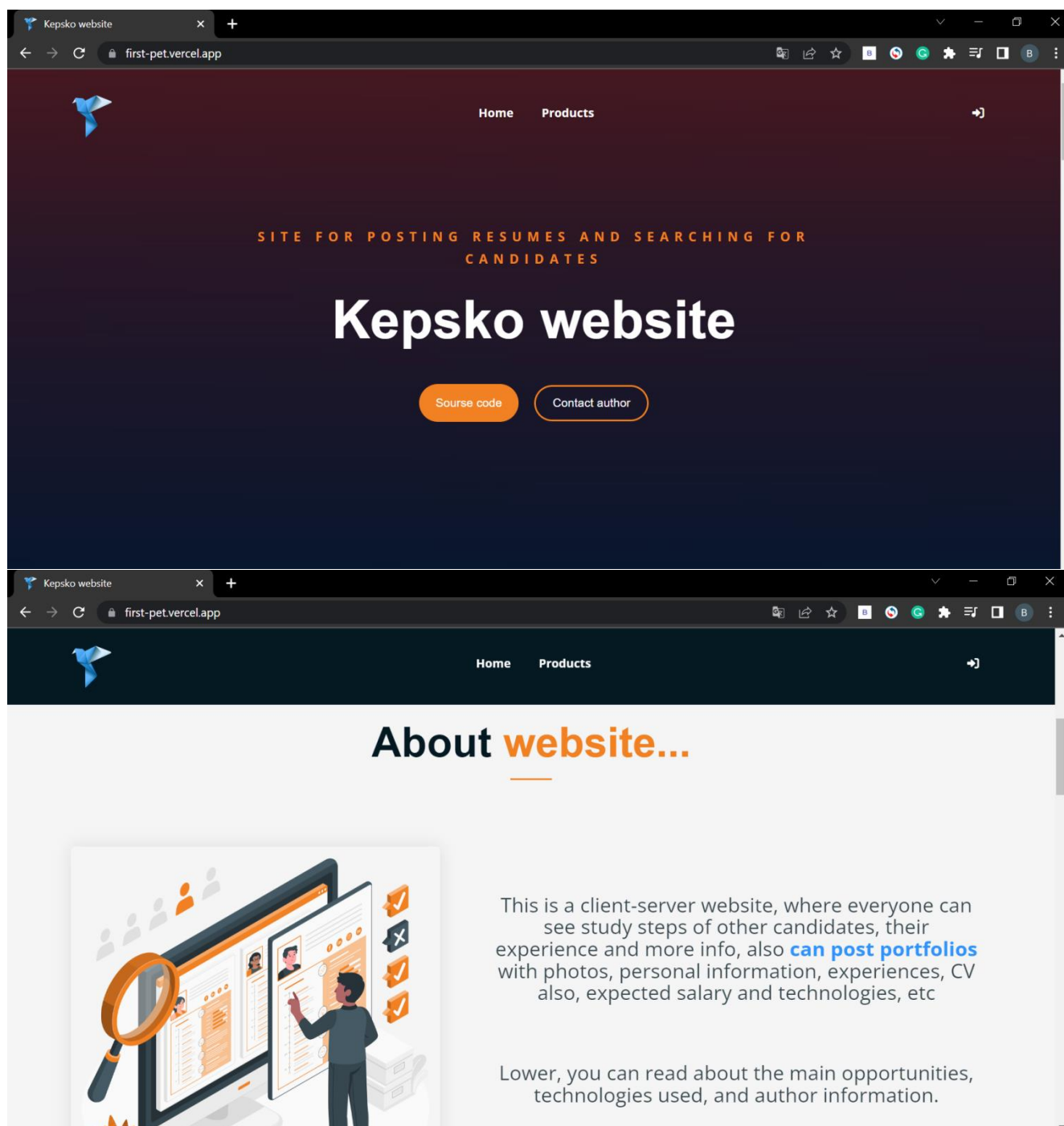
1. *Bootstrap*: офіційна онлайн документація для розробки швидких та красивих вебсайтів [Електронний ресурс] – Режим доступу до ресурсу: <https://getbootstrap.com/>
2. *Node.js*: open-source, cross-platform JavaScript runtime environment [Електронний ресурс] – Режим доступу до ресурсу: <https://nodejs.org/en>
3. *MongoDB*: The Developer Data Platform [Електронний ресурс] – Режим доступу до ресурсу: <https://www.mongodb.com/>
4. *SendinBlue*: [Електронний ресурс] – Режим доступу до ресурсу: <https://www.brevo.com/products/transactional-email/>
5. *TinyMCE*: The Most Advanced WYSIWYG Editor [Електронний ресурс] – Режим доступу до ресурсу: <https://www.tiny.cloud/>
6. *GoogleAuth*: Quickly build and deploy Node.js applications on Google Cloud. Документація для Node.js [Електронний ресурс] – Режим доступу до ресурсу: <https://cloud.google.com/nodejs/docs/reference/google-auth-library/latest/google-auth-library/googleauth>
7. *Vercel*: platform for frontend and backend developers, providing the speed and reliability innovators need to create at the moment of inspiration [Електронний ресурс] – Режим доступу до ресурсу: <https://vercel.com/>
8. *Render*: Cloud Application Hosting for Developers [Електронний ресурс] – Режим доступу до ресурсу: <https://render.com/>
9. *Visual Studio Code* - The best IDE for JavaScript coding [Електронний ресурс] – Режим доступу до ресурсу: <https://code.visualstudio.com/>
10. *Postman*: API platform for building and using APIs [Електронний ресурс] – Режим доступу до ресурсу: <https://www.postman.com/>
11. *GitHub*: Internet hosting service for software development and version control using Git [Електронний ресурс] – Режим доступу до ресурсу: <https://github.com/>

12. *Djinni*: Platform, not an agency for hiring candidates [Електронний ресурс] – Режим доступу до ресурсу: <https://djinni.co/>
13. *Glassdoor*: American website where current and former employees anonymously review companies [Електронний ресурс] – Режим доступу до ресурсу: <https://www.glassdoor.com/index.htm>
14. *Robota.ua*: Український вебсайт для пошуку роботи [Електронний ресурс] – Режим доступу до ресурсу: <https://rabota.ua/ua>
15. *Pracuj.pl* - Польський вебсайт для пошуку роботи [Електронний ресурс] – Режим доступу до ресурсу: <https://www.pracuj.pl/>
16. *Express.js*: minimalist web framework for Node.js [Електронний ресурс] – Режим доступу до ресурсу: <https://expressjs.com/>
17. *Райан Дал* — програмний інженер і перший розробник проєкту Node.js. [Електронний ресурс] – Режим доступу до ресурсу: <https://github.com/ry>
18. *Українська документація Node.js*: [Електронний ресурс] – Режим доступу до ресурсу: <https://nodejs.org/uk/docs>
19. *NPM*: library and registry for JavaScript software packages [Електронний ресурс] – Режим доступу до ресурсу: <https://www.npmjs.com/>
20. *EJS*: Embedded JavaScript templating [Електронний ресурс] – Режим доступу до ресурсу: <https://ejs.co/>
21. *Bootstrap Themes*: [Електронний ресурс] – Режим доступу до ресурсу: <https://themes.getbootstrap.com/>
22. *Bootstrap Icons*: [Електронний ресурс] – Режим доступу до ресурсу: <https://icons.getbootstrap.com/>
23. *Figma* — графічний онлайн-редактор для спільної роботи [Електронний ресурс] – Режим доступу до ресурсу: <https://www.figma.com/>
24. *Bootstrap Examples*: розроблені програмою приклади сторінок [Електронний ресурс] – Режим доступу до ресурсу: <https://getbootstrap.com/docs/5.3/examples/>

25. *Brevo* — platform that helps to build customer relationships across email, SMS, chat, and more [Электронный ресурс] – Режим доступа до ресурсу: <https://www.brevo.com/>
26. *Nodemailer*: module for Node.js applications to allow easy as cake email sending. [Электронный ресурс] – Режим доступа до ресурсу: <https://nodemailer.com/about/>
27. *Mailgun*: Transactional Email API Service For Developers [Электронный ресурс] – Режим доступа до ресурсу: <https://www.mailgun.com/>
28. *SendGrid*: Email Delivery, API, Marketing Service [Электронный ресурс] – Режим доступа до ресурсу: <https://sendgrid.com/>
29. *OAuth2.0*: the industry-standard protocol for authorization [Электронный ресурс] – Режим доступа до ресурсу: <https://oauth.net/2/>
30. *Google Cloud*: Cloud Computing Services [Электронный ресурс] – Режим доступа до ресурсу: <https://cloud.google.com/>
31. *Heroku*: Cloud Application Platform [Электронный ресурс] – Режим доступа до ресурсу: <https://www.heroku.com/>
32. *Next.js* by Vercel - The React Framework for the Web [Электронный ресурс] – Режим доступа до ресурсу: <https://nextjs.org/>
33. *Source code*: [Электронный ресурс] – Режим доступа до ресурсу: https://github.com/vasy11312/first_pet
34. *CryptoJS*: growing collection of standard and secure cryptographic algorithms implemented in JavaScript using best practices and patterns. [Электронный ресурс] – Режим доступа до ресурсу: <https://code.google.com/archive/p/crypto-js/>
35. D. Flanagan, “JavaScript: The Definitive Guide, 7th Edition” – 2020.
36. G. Lim, “Beginning Node.js, Express & MongoDB Development” – 2019.

ДОДАТКИ

Додаток А.1 Вигляд основної сторінки вебсайту для незареєстрованих користувачів



Kepsko website first-pet.vercel.app

Home Products


Main opportunities

Look up a candidate
Here, everyone can **search for a candidate** depending on their requirements. Other clients can read info about the person, their study steps, an example of pet projects, and more...

Communication
When you found a good resume, you can add him to your **favorites**, and after this - **send a letter** to the owner.

Add new resume
You can **add** your own resume with avatar, expected salary, **categories**, and description where you can add pet projects, study steps, **more information** about soft and hard skills, and more...

Log In / Sign Up
Everyone can **register** his account or **log in** there also using Google Auth. Users can update their **profiles** or **edit/delete their own resumes**.



Kepsko website first-pet.vercel.app

Home Products

Used technologies

Backend side
Used **Node.js** here for main tasks and logic.

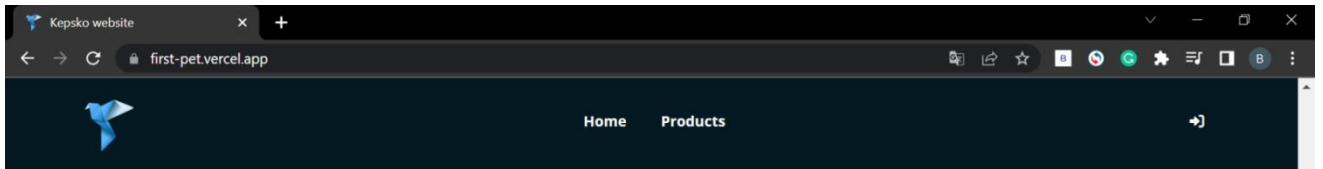
Database
MongoDB was used to organize and store data for future processing, retrieval, or evaluation.

Frontend side
Visualization was used for **EJS** and **Bootstrap**. Also, for styles using **CSS**.

Server
The server side used **Express.js**. **RESTful API** for client-server interaction, which used **HTTP methods**.

Sending email
When the user tries to **sign up**, **reset the password**, **contact the candidate**, etc - send an email using **SendinBlue**.

OAuth
The website has **login and registration** methods, also with **GoogleAuth** to more comfortable to sign in for users.



About Author

Here you can read info about the author of this website, CV, social media, and photo.

VASYL HRYTS BACKEND DEVELOPER

CONTACT

Address :
Vyshneve, Kyiv region,
Ukraine

Mobile :
+380 (98) 711-73-72

Email :
vasylhryts74@gmail.com

Telegram :
[t.me/vasylhryts](#)

GitHub :
<https://github.com/vasylhryts>

EDUCATION

TARAS SHEVCHENKO NATIONAL UNIVERSITY OF KYIV
BACHELOR OF COMPUTER SCIENCES AND CYBERNETICS
2019 — 2023

SKILLS

- Node.js
- Express
- MongoDB
- JavaScript
- REST API
- HTML, CSS, EJS
- Git, GitHub
- SendGrid/SendinBlue
- React.js

LANGUAGES

English	B1
Ukrainian	C2
D. russian	r1
Russian	L1

EXPERIENCES

UNIVERSITY INTERNSHIP || SEPTEMBER-DECEMBER 2022

As a **Backend (Node.js) Developer** I was in charge of developing **a client-server website** where everyone can post portfolios with photos, personal information, experiences, etc.

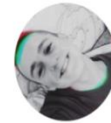
- **authorization** on web-site also using OAuth2;
- **database interaction (MongoDB)**: using HTTP Methods for RESTful API Services;
- **users communication**: used SendinBlue; was made communication history page;
- **password recovery**: sending a confirmation link to the email using a temporary token;
- **filtering** portfolios due to their skills;

Currently, I am working on an upgrade of this website and the following tools will be added:

- **more information** can be added to resumes: pet projects, video-CV, study steps, etc;
- **chatbot** for user's communication through the website;
- **user-admin** with permission to moderate spam;
- **and other** features you can see on my GitHub.

UNIVERSITY TEAMWORK || FEBRUARY-APRIL 2023

Also, I have a **team project** where I am in charge of **Frontend(React)**. It's an excellent experience to work with a team. And will broaden my mind in Frontend development.



Hryts Vasyl

Backend developer

I am a student studying Cybernetics and Computer Science at the [Taras Shevchenko National University of Kyiv](#).

Motivated Node.js enthusiast seeking an offer to a **Backend position as a developer**. Possess a strong foundation in computer science concepts and hands-on experience in JS programming. Committed to continuous learning and improvement in the field.

This website started as university internship work, after as graduate work. More info, you can read in my

CV

My social media:



Додаток А.2 Сторінка реєстрації користувачів

Kepsko website

first-pet.vercel.app/register

Sign up to continue

Nickname

Email

Password

Already have an account?

SIGN UP

- or sign up using -

SIGN UP WITH GOOGLE

Додаток А.3 Сторінка авторизації користувачів

Kepsko website

first-pet.vercel.app/login

Login to continue

Email

Password

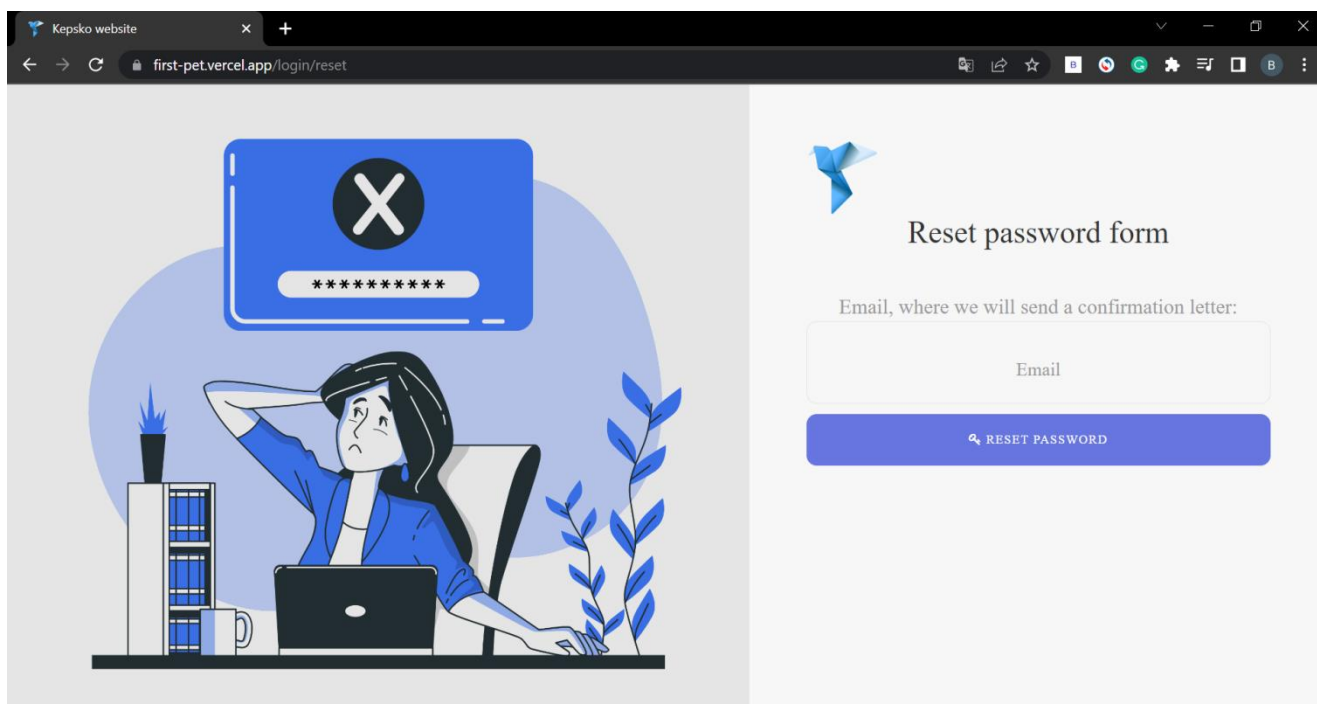
Don't have an account? Forgot Password?

LOGIN

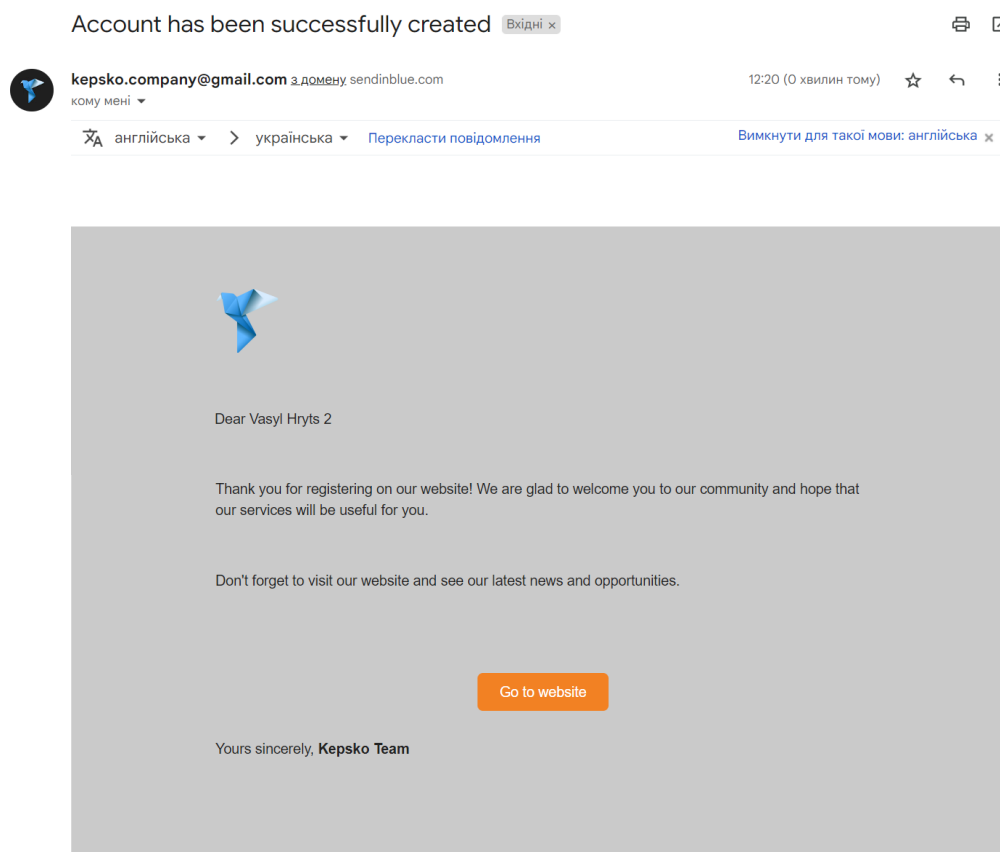
- or sign up using -

LOGIN WITH GOOGLE

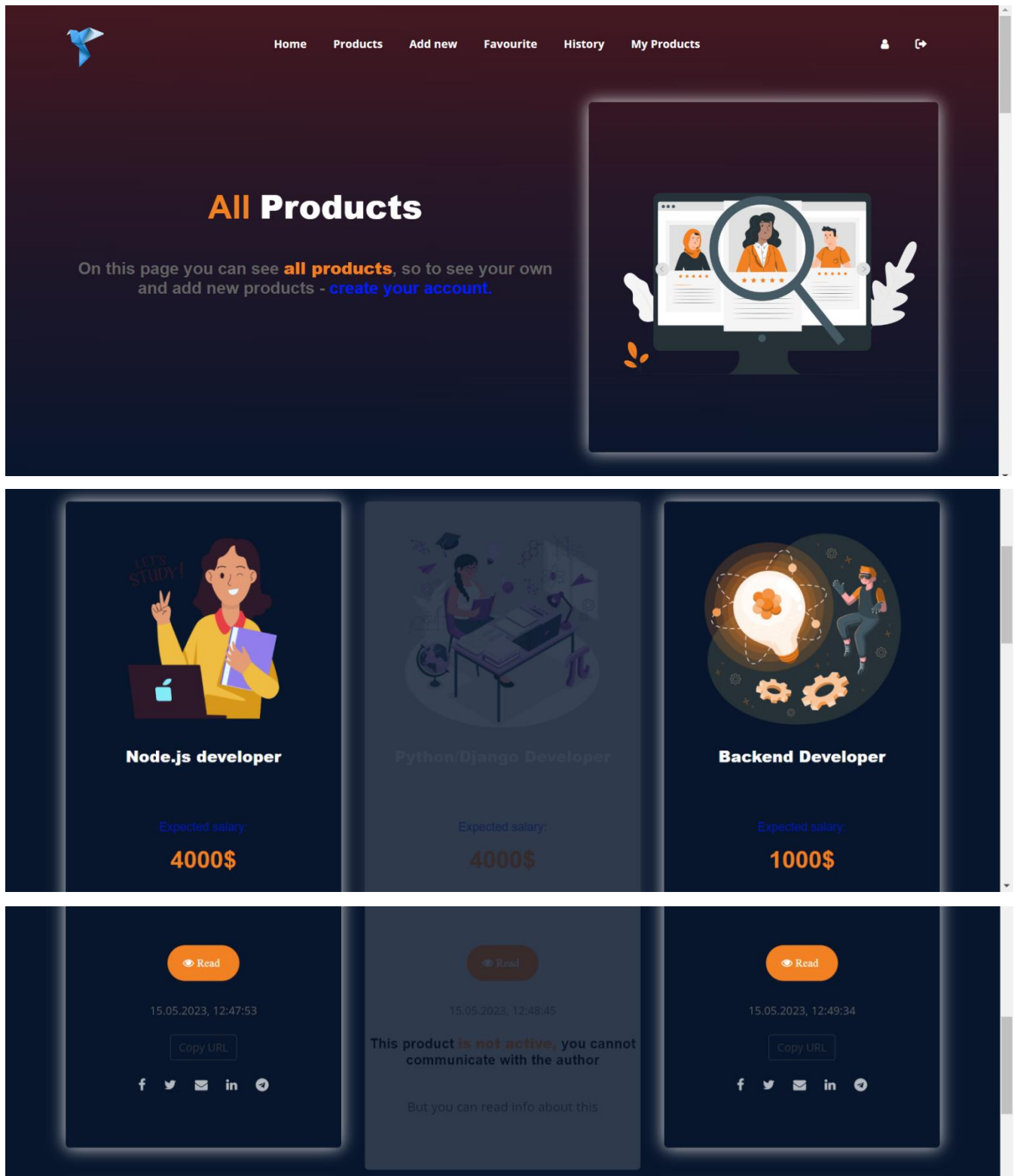
Додаток А.4 Сторінка відновлення паролю



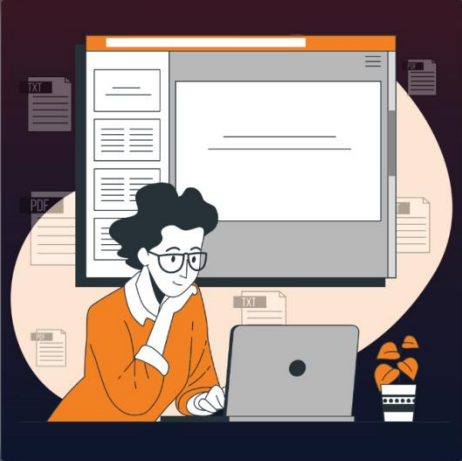
Додаток А.5 Приклад електронного листа після створення акаунту



Додаток Б Сторінка усіх резюме з додатковими можливостями на верхній панелі для зареєстрованих користувачів



Додаток В Сторінка окремого резюме



Home Products Add new Favourite History My Products


Read Product

On this page, you can see **info about this product**, you can to add this product to favourite and add to favourite

Backend Developer

Expected salary:

1000\$



File Edit View Insert Format

Paragraph B I

Студент який шукає роботу з нахилом на Node.js

Освіта	Студент 4курсу КНУ на факультеті Комп'ютерних наук та Кібернетики
Курси	<ul style="list-style-type: none"> html/css знайомство і курс, знайомство з фронтеда на js книга по js курс по js з практикою Знайомство з node.js та Express

tiny

Categories: javascript backend

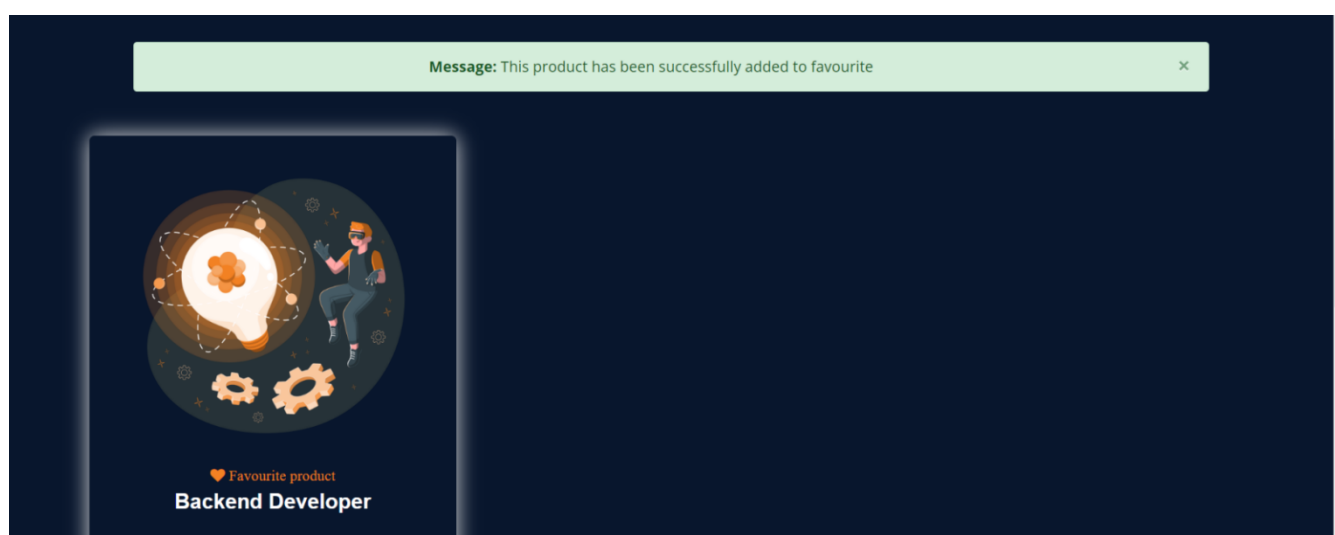
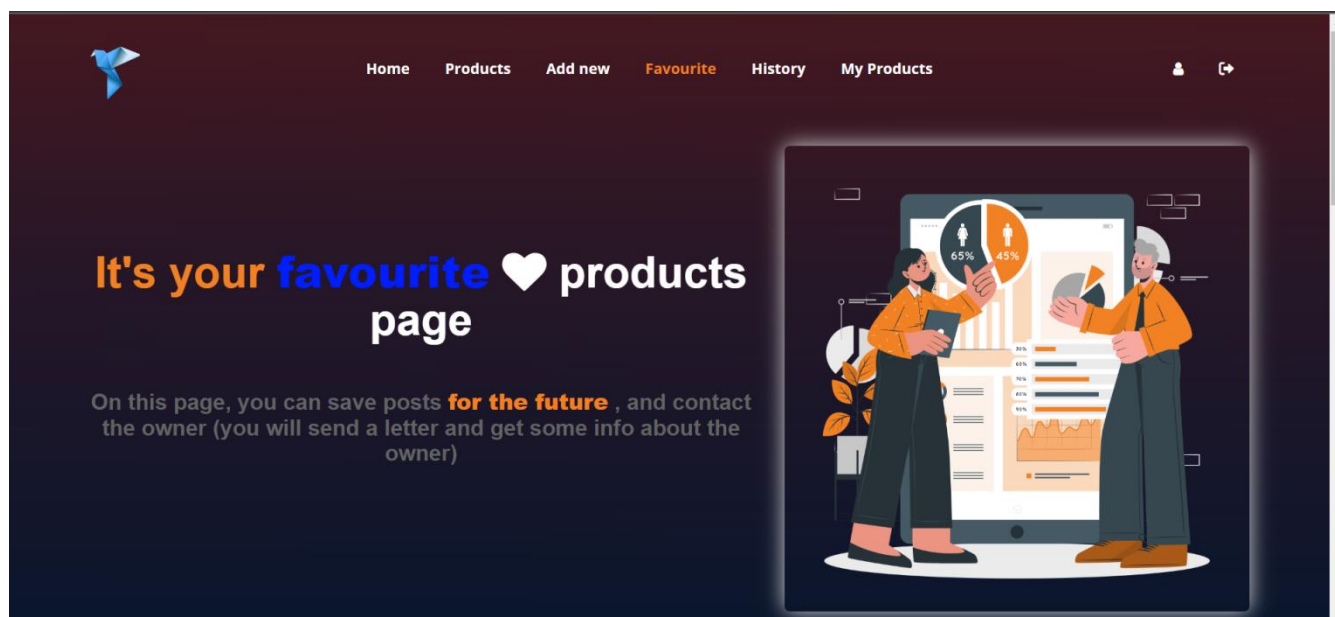
Share: Copy URL

f t e in

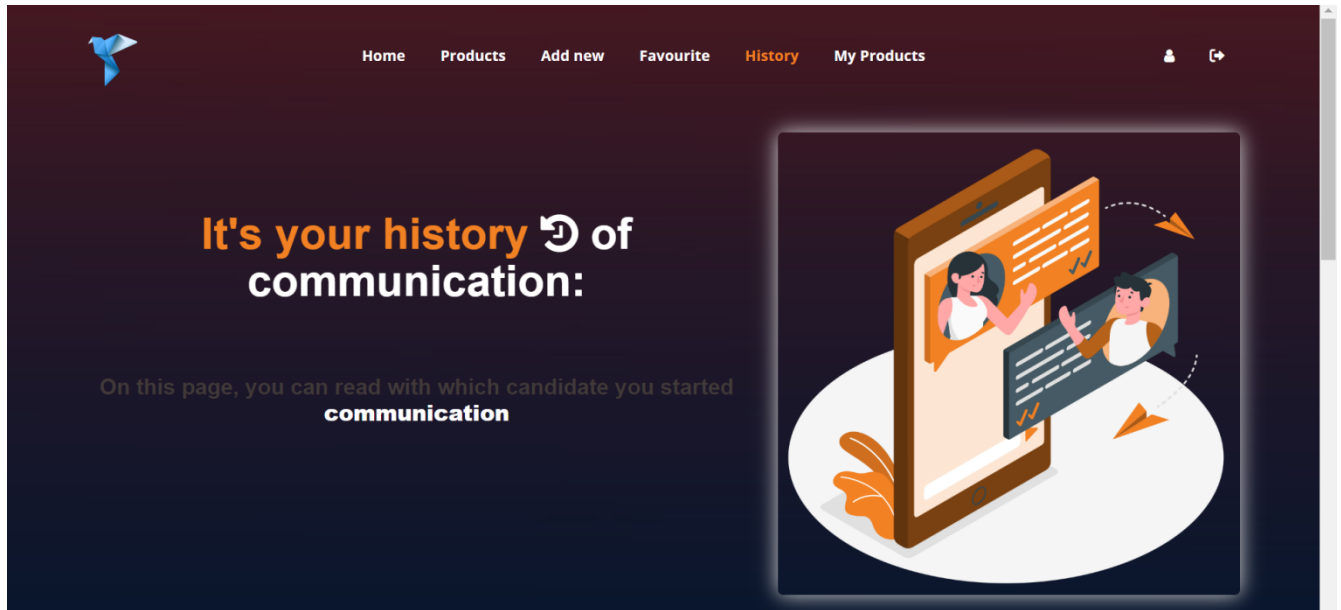
15.05.2023, 12:49:34

Add to favourite

Додаток Г Сторінка улюблених резюме



Додаток Г.1 Сторінка історій комунікації



Message: You successfully sent mail to this person

Product title	Owner Name	Owner Email	Expected salary	Date, when message was sent	Delete
Python Developer	mexedi dabro	dabromexedi@gmail.com	2000\$	21.03.2023 19:03:46	
Backend Developer	mexedi dabro	dabromexedi@gmail.com	1000\$	19.05.2023 12:54:13	

Додаток Г.2 Приклад електронного листа після початку комунікації

Hello, Your portfolio is interesting to me) Вхідні x



kepsko.company@gmail.com з домену sendinblue.com
кому мені ▾

12:54 (2 хвилини тому)



Hi, mexedi dabro, my email is: vasylhryts@knu.ua

Your portfolio:

Backend Developer

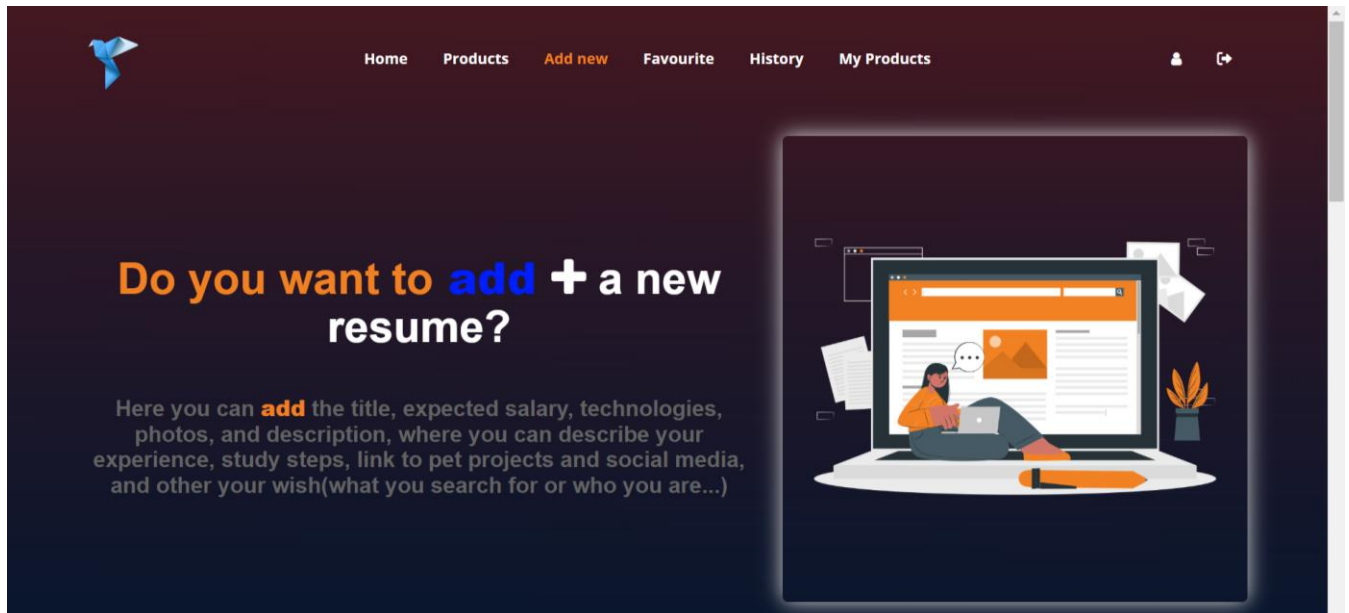
is so interesting for me,
We can communicate about studying or other here

Please, let me know if you got this letter.

[Go to website](#)

Yours sincerely, **Kepsko Team**

Додаток Д.1 Сторінка створення резюме



Please enter your speciality

Enter expected salary in \$ (value must be integer)

File Edit View Insert Format Tools Table Help

Paragraph B I [Text alignment icons]

Description, here you can add more information about yourself, your projects, steps of study, links, and more...

Please select categories

Start write...

Please select photo of product (jpg/png/jpeg)

Вибрати файл файл не вибрано

Add new +

Додаток Д.2 Сторінка створення резюме та демонстрація додаткових можливостей опису інформації

The screenshot displays the TinyMCE Cloud demo page within a web browser. The page features a dark blue header with navigation links: Home, Products, Add new, Favourite, History, and My Products. The main content area is a rich text editor showing the following elements:

- A welcome message: **Welcome to the TinyMCE Cloud demo!**
- A section titled **Got questions or need help?** with a list of links:
 - Our [documentation](#) is a great resource for learning how to configure TinyMCE.
 - Have a specific question? Try the [tiny_mce_tag at Stack Overflow](#).
 - We also offer enterprise-grade support as part of [TinyMCE premium subscriptions](#).
- A section titled **A simple table to play with** containing a table:

Product	Cost	Really?
TinyMCE Cloud	Get started for free	YES!
Plupload	Free	YES!
- A section titled **A video demonstration** showing a video player with a red play button and a YouTube logo.

At the bottom of the page, there is a form with the text "Please select categories" and a dropdown menu with the following options: javascript, java, python, c_similar, frontend, backend, web_design, and other. The page also shows a word count of 66 words and a TinyMCE logo.

Додаток Е Сторінка усіх резюме, створених автором

Home Products Add new Favourite History My Products

My products ✓

On this page, you can see **only your products**, update and delete them

✓ Your product

Java Beginner

Expected salary

300\$

Edit

19.05.2023, 13:39:07

Copy URL

f t e in w

Додаток Є Редагування вибраного резюме



Here you can **edit** the title, expected salary, technologies, photos, description and more...

Title:

Java Beginner

Expected salary:

300

Description:

File Edit View Insert Format Tools Table Help

Heading 2 B I

Hello, I'm new to programming 🧑🎓

About me?

- My first pet project is in this [documentation](#)
- Am a student, so want to be a good programmer

Product	Cost	Still, updating?
First	free	YES!
Course project	100\$	YES!

h2 39 words tiny

Categories: java c_similar other

Please select **new categories**

Please select **another image** (jpg/png/jpeg)

файл не вибрано

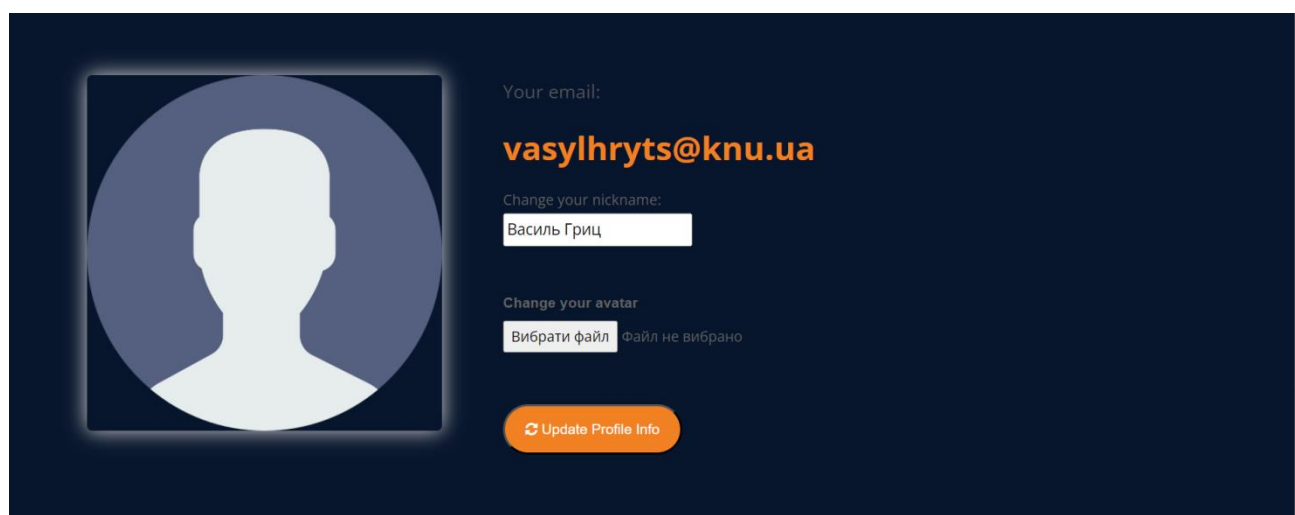
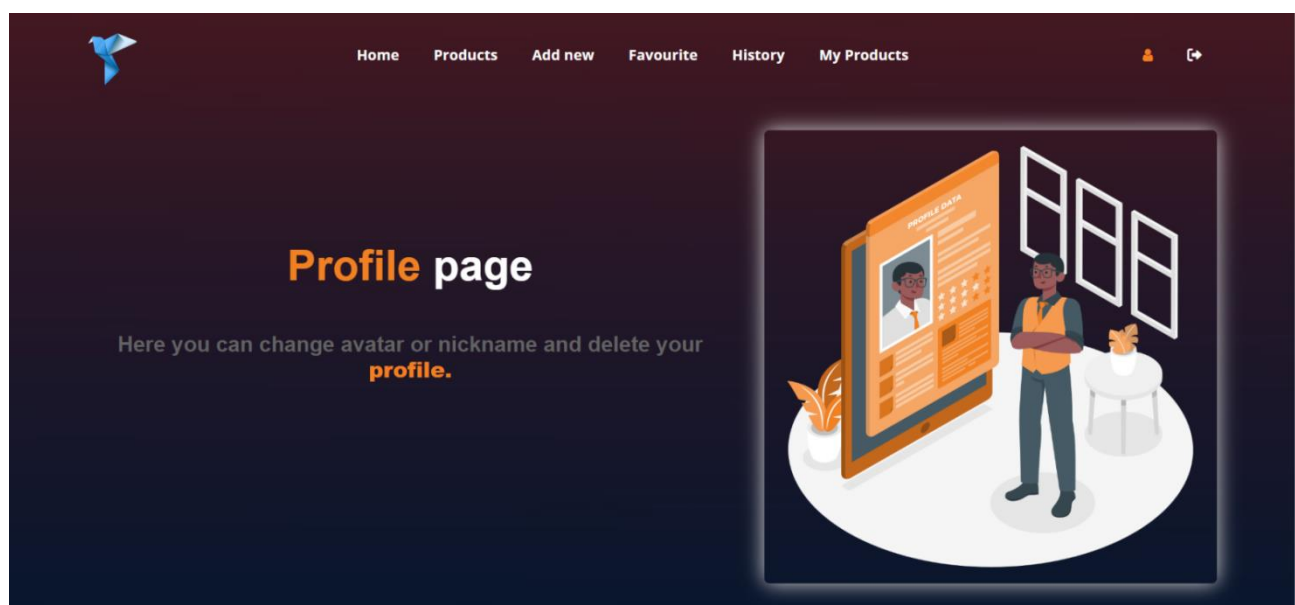
Your product will **active**, do you want to make it passive(everyone will read your

resume but is not able to contact you)?



Or

Додаток Ж Сторінка профілю користувача



Додаток 3 Сторінка певної категорії

The screenshot displays a web application interface for a category page. At the top, there is a navigation bar with links: Home, Products, Add new, Favourite, History, and My Products. A user profile icon and a share icon are also present. The main content area has a dark background with the following elements:

- Category Header:** "Category: Java" in large orange text, followed by a sub-header: "On this page, you can see all filtered products with the category **Java**".
- Illustration:** A large illustration of a person sitting at a desk with a laptop, with a large screen showing a website layout.
- Product Cards:** Two cards are displayed side-by-side:
 - JAVA Backend:** Features an illustration of a person at a laptop with code snippets (CSS, c/>) and gears. The price is **2000\$**. Below the price is a "Read" button, a timestamp "15.05.2023, 12:47:04", a "Copy URL" button, and social media icons (Facebook, Twitter, Email, LinkedIn, WhatsApp).
 - Java Beginner:** Features an illustration of a person holding a lantern. The price is **300\$**. Below the price is a "Read" button, a timestamp "19.05.2023, 13:39:07", a "Copy URL" button, and social media icons.