COMP3030J Software Engineering Project 2 - 2024-2025
Beijing-Dublin International College

COMP3030J Software Engineering Project 2

System Document

Students:
Yani Yang- 22207291
Qiyue Zhu- 22207232
Zihan Miao- 22207265
Wenyi Liang- 22207277
Boran Duan- 22207281

Professor:
Dr. Catherine Mooney

Group 4: QUEENs NEVER CRY

May 30, 2025

Contents

1 Introduction

1.1 Project Overview

QUEENs NEVER CRY is a purpose-driven company consisting of 5 queens, dedicated to empowering vulnerable women through our innovative digital platform. Developed in close collaboration with our client, **the Women's Federation**, our platform is a trusted sanctuary where women can access essential resources, share information, and explore diverse employment opportunities.

Our primary clients include organizations and institutions actively promoting women's well-being and professional growth. By cultivating an inclusive and supportive community, we equip women with the tools and knowledge they need to overcome challenges, enhance their confidence, and realize their aspirations. Together, we are on a mission to create a safer, more equitable future for all women.

1.2 Problem Statement

The lack of investment in women and gender equality is not only a social issue but also a significant economic burden, costing the global economy $10 trillion annually. Additionally, more than 850 million women and girls in countries with poor gender equality face numerous challenges, including forced pregnancy, child marriage, and limited access to education. Despite the United Nations Sustainable Development Goals (SDGs) highlighting Gender Equality as a crucial objective, impoverished women continue to struggle with limited educational and employment opportunities.

In the current market environment, the Women's Federation website is primarily used for displaying meeting agendas and ideological reports, lacking a national-level, comprehensive women and children assistance website. This market situation has led to various issues, such as the following: the lack of educational and job opportunities, insufficient communication and support platforms, scarcity of educational resources, limited employment and career development channels, and inefficient operational management. These challenges restrict the personal and professional growth of women and children, hindering their ability to share experiences, seek support, access reliable guidance, connect to job opportunities.

In response to these pressing challenges, our project aims to establish a website dedicated to empowering impoverished women by providing access to education and employment opportunities. The project aligns with SDGs Goal 1 (No Poverty), Goal 4 (Quality Education), Goal 5 (Gender Equality), and Goal 8 (Decent Work and Economic Growth), emphasizing the urgent need to empower women through educational and employment support initiatives.

1.3 Vision Statement

Our vision is to create a digital platform that goes beyond traditional job-matching tools, offering tailored support and real opportunities for vulnerable women. Developed in close partnership with the Women's Federation, the platform combines user insights, shared resources, and targeted employment services to help women build confidence and achieve long-term growth.

We aim not only to enhance women's economic independence but also to drive broader progress toward gender equality. This vision reflects our commitment to social justice and supports the United Nations Sustainable Development Goal of Gender Equality.

Our platform will provide marginalized women with access to educational resources, job opportunities, and support networks, empowering them to achieve personal and professional growth. By bridging the access gap, we help women enhance their skills, gain employment, and work toward long-term self-sufficiency. Additionally, our platform fosters community connections, offering emotional support and mentorship to reduce isolation and build resilience. With accessible information on legal rights, mental health, and career readiness, women can make informed decisions and advocate for themselves. Partner organizations will also benefit from tools to manage content, track engagement, and measure their impact, ensuring more effective support and scalable efforts.

Collectively, our project addresses the immediate and long-term needs of marginalized women. By solving these challenges, we aim to fill critical gaps in the market, promote gender equality, and contribute to sustainable development goals. Ultimately, our platform envisions a future where women are empowered, communities are enriched, and society as a whole progresses toward greater equality and opportunity for all.

1.4 Sustainable Development Goals Analysis

- **SDG 1: No Poverty** Our project aims to empower vulnerable women by providing them with access to essential resources, information sharing, and diverse employment opportunities through our innovative digital platform. By fostering an inclusive and supportive community, we equip women with the tools and knowledge they need to overcome challenges, enhance confidence, and realize their aspirations. This mission contributes to reducing poverty and providing more economic opportunities for women, thus contributing to the achievement of the No Poverty goal.
- **SDG 4: Quality Education** Through our digital platform, women can access educational resources and training to enhance their skills and knowledge, promoting self-development and career growth. Our project provides convenient pathways for training and education, contributing to the goal of quality education and creating more learning opportunities for women.
- **SDG 5: Gender Equality** Our project promotes gender equality by empowering women, providing equal employment opportunities and educational resources. We break traditional barriers, enabling women to participate in economic and social activities and advancing the gender equality agenda. By supporting women's professional growth and self-realization, we strive towards achieving the goal of gender equality.
- **SDG 8: Decent Work And Economic Growth** Our project offers women decent and diverse job opportunities, helping them integrate into the labor market and achieve economic independence. By establishing a sustainable digital platform, we promote economic growth and drive women's participation in various industries, contributing to the goal of decent work and economic growth.

1.5 Project Methodology

Our project will implement the agile methodology, a dynamic approach that values flexibility, iterative progress, and collaboration with stakeholders. In contrast to traditional waterfall methods, this approach enables quick responses to changes and promotes continuous improvement through regular reflection and adjustments. The agile approach is well-suited for our project as it can adapt to evolving circumstances and requirements. It fosters a customer-centric approach, allowing us to swiftly and effectively incorporate stakeholder feedback into the development process. This iterative process ensures that the project aligns with stakeholder expectations and enhances the final output quality through ongoing refinement.
To further illustrate our project management strategy, we have developed a Gantt chart, depicted in Figure 1. This chart outlines a detailed timeline and sequence of activities, showcasing our thorough planning and unwavering commitment to adaptability.

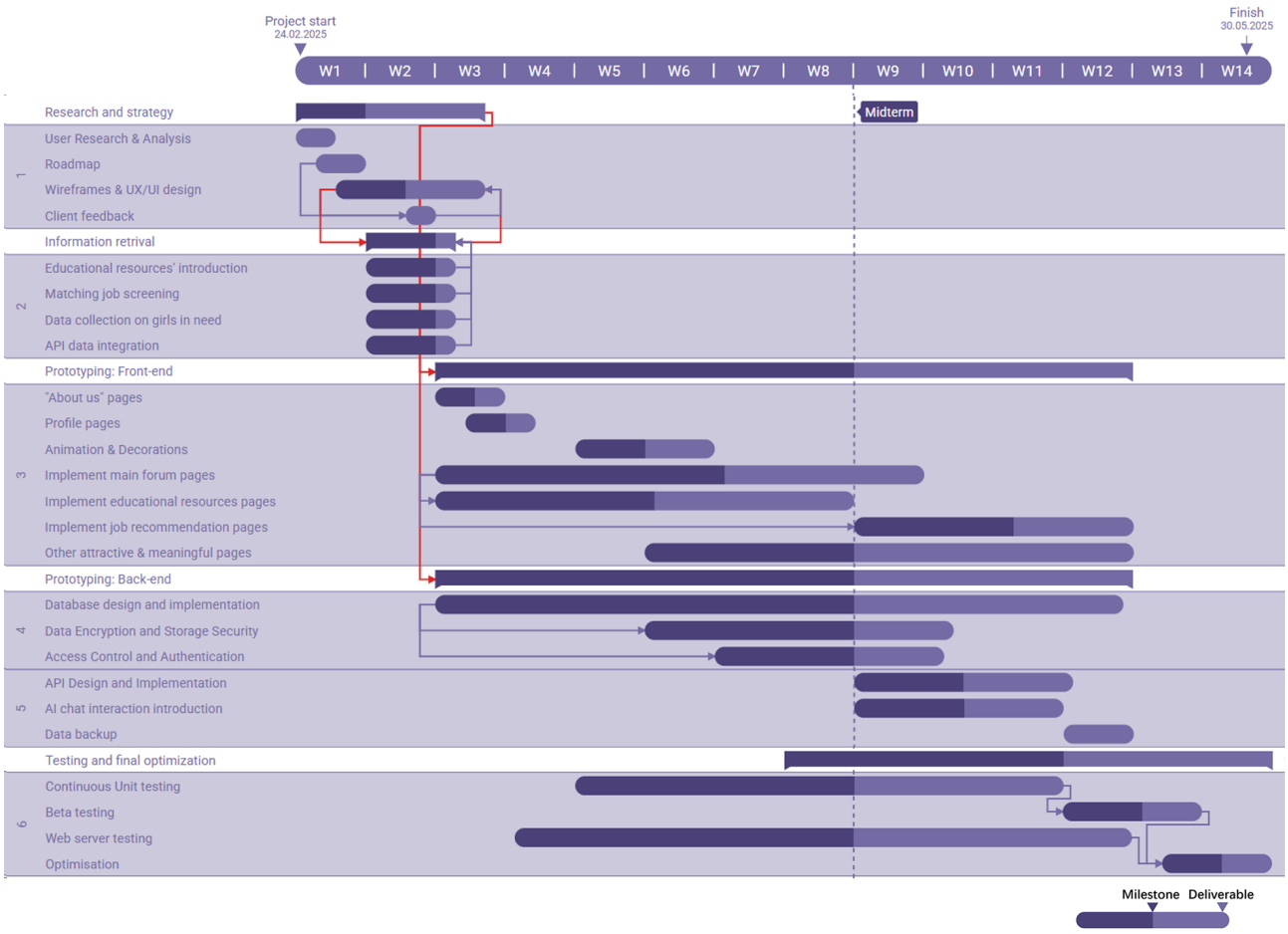

Figure 1: Gantt Graph

2 Group Work

2.1 Member Contributions

Each member of our team made equal contributions to the project. The specific responsibilities of each individual are detailed in Table 1. Each individual's unique skill set and expertise contributed to the project's overall success, with everyone playing a crucial role in their respective areas.

Table 1: Group Member Responsibilities and Contributions

Group Member	**Responsibilities**	**Contribution (%)**
Yani Yang	Visionary leader & Strategic planner	20%
Qiyue Zhu	UI Design & Front-end Development	20%
Zihan Miao	UX Design & Front-end Development	20%
Wenyi Liang	Testing & Front-end Development	20%
Boran Duan	Back end Development & Database	20%
Project Initiation	Professors and Financial Endorsement	-
	Teaching Assistants Resource Provision	-
	Project Supervision	-
	Project Sponsors	-

2.2 Collaborative Development

During the development process, our team created the organization COMP3030J-Software-Engineering-Project-2 on the GitHub platform. All team members and our Teaching Assistant were included in this organization.

GitHub was used as the main platform for collaborative development, facilitating smooth coordination and version control of project resources.

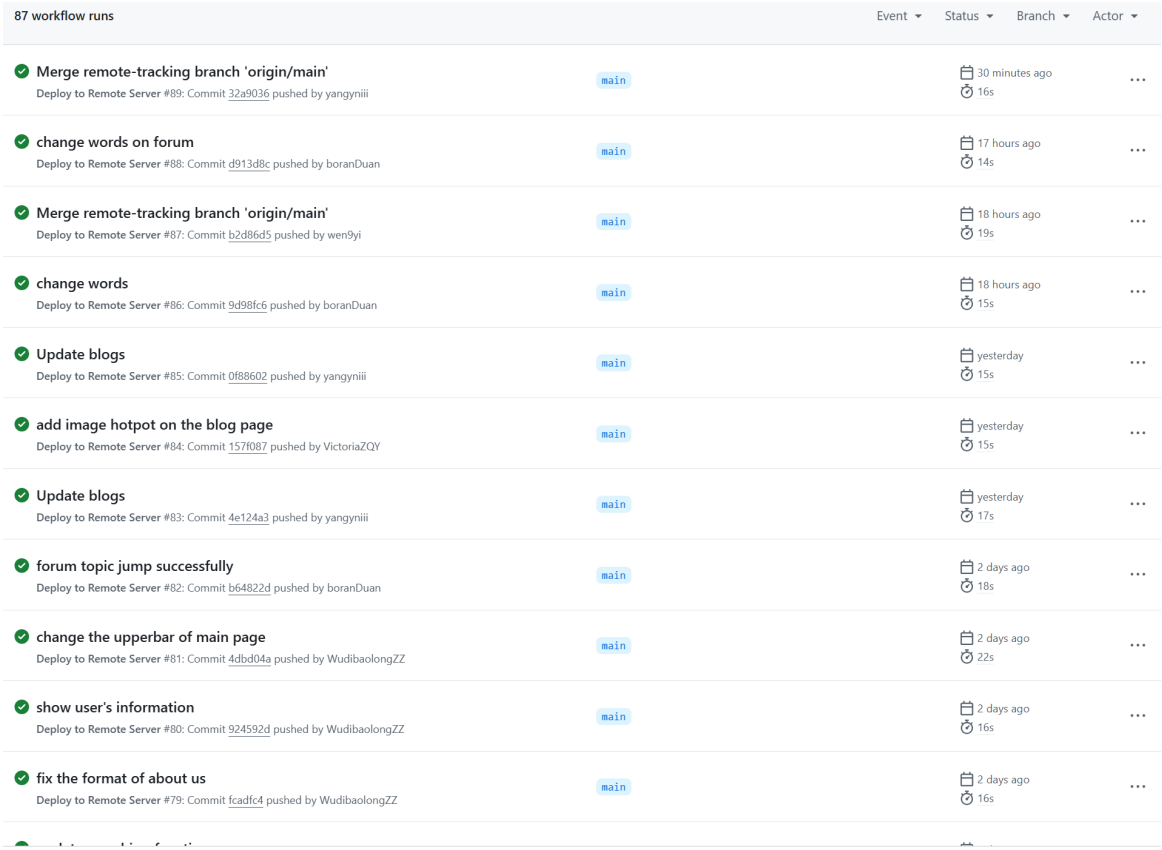

Figure 2: Github Workflow

We have streamlined our collaboration and task tracking by using GitHub Log (Figure 3) to clearly document individual contributions. Each team member consistently committed their code with descriptive messages outlining the tasks they completed. This approach not only provided transparency regarding who was responsible for which parts of the project but also enabled us to trace the development timeline with clarity. The disciplined use of GitHub Log fostered accountability and facilitated efficient coordination throughout the project lifecycle.

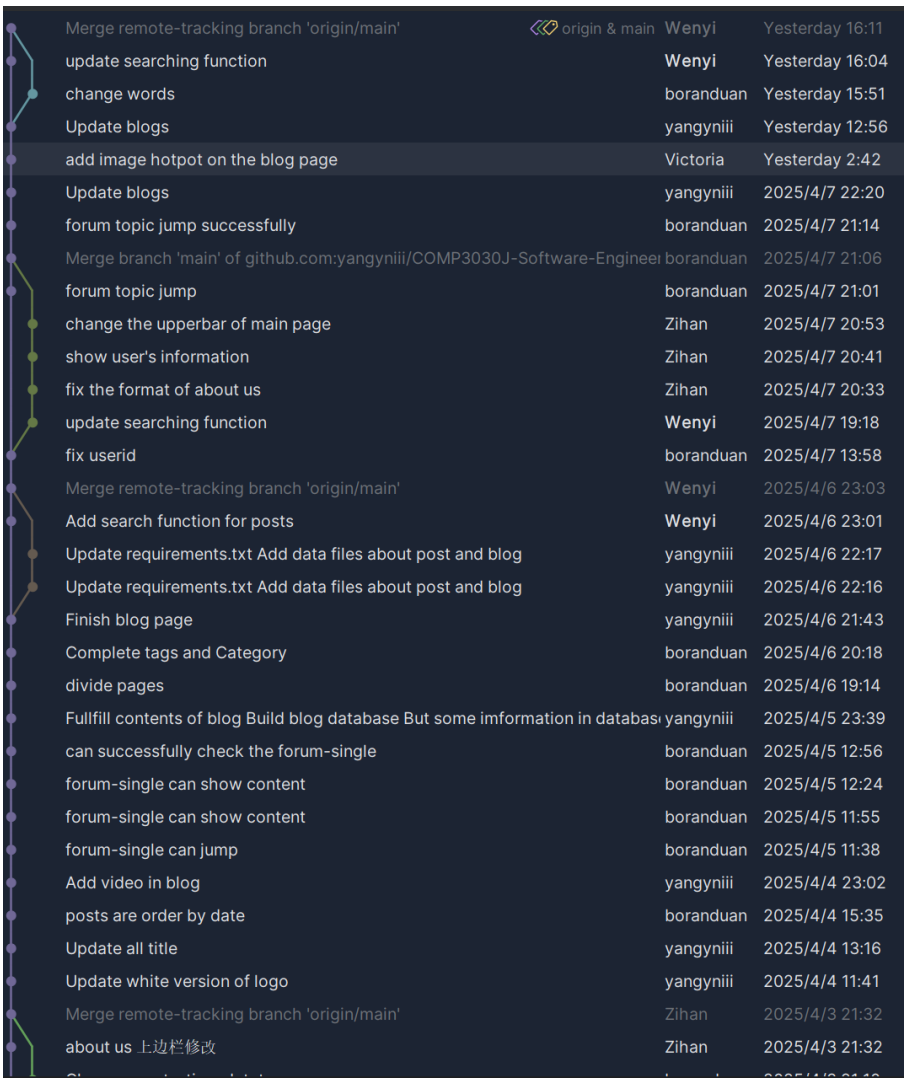

Figure 3: Github Log

Simultaneously with project development, our team collaboratively created developer documentation and various API-related documents. These resources are stored in the /doc directory of our repository. Presently, user documentation serves as a guide for future developers, enhancing project sustainability by offering insights into our development process and problem-solving approaches.

In addition to GitHub, we have established a WeChat group chat to facilitate real-time communication and collaboration. This platform has proven essential in cultivating a dynamic and responsive team environment. Instant communication has streamlined interactions, allowing us to promptly address queries, exchange ideas, and make collective decisions.

We have used Overleaf, a collaborative LaTeX editor, for writing and editing our project documentation. The real-time collaboration features of Overleaf have enabled us to work simultaneously on documents, enhancing efficiency and ensuring consistency in our written materials. This platform also offers robust version control, ensuring all changes are tracked and that the latest version of a document is always accessible.

By strategically leveraging these collaborative development tools, we have cultivated a highly efficient and communicative team environment. These tools have boosted our productivity and ensured adherence to high quality and consistency standards throughout the project's development cycle.

2.3 Team Conflict

Throughout the development journey of our project, we encountered various conflicts that required proactive resolution and collaborative problem-solving. Drawing upon the principles discussed in our classes on "How to manage group conflicts," we confronted these challenges directly, nurturing a culture of transparent communication and teamwork.

A significant conflict emerged in the early stages of development stemming from divergent opinions on selecting frameworks and the technology stack. With each team member bringing unique skills and preferences, discussions and disagreements ensued on the best approach. To address this, we employed multiple communication avenues, including face-to-face meetings, a WeChat group, and GitHub's discussion platform. Through comprehensive deliberations, we assessed the pros and cons of different frameworks and technologies. All crucial decisions and ideas were documented on GitHub discussions, ensuring transparency and encouraging every team member to contribute their perspectives. This collaborative process led us to a consensus on the most fitting technology stack, taking into account project requirements and team members' expertise.

Throughout our development journey, several minor conflicts of a similar nature were encountered. Nonetheless, each conflict was effectively resolved through open communication, mutual understanding, and a dedication to achieving our common project objectives.

Another challenge emerged when we faced issues handling high forum traffic and active discussions on the platform. After assessing the situation, we consulted with our technical advisor to better understand the potential bottlenecks. Based on their recommendations, we decided to implement scalable cloud hosting solutions and optimize database indexing. While these improvements were underway, we carefully monitored system performance to ensure that users continued to experience smooth interactions, even during peak usage periods.

By actively addressing conflicts and challenges as they surfaced, we upheld team progress and unity. Our readiness to communicate transparently, seek support when necessary, and take proactive measures in finding solutions empowered us to overcome hurdles and move forward with the project.

3 System Design

This portion outlines the system architecture of our project, emphasizing the key components and technologies employed. The system design is essential for the effective realization of our project goals and specifications.

3.1 Technology Stack

Our system is built on a robust technology stack comprising various frameworks, tools, and services that support the development and functionality of our project. It is crucial to emphasize that every technology and tool integrated into our stack underwent thorough team deliberations and received unanimous approval from all team members. Figure 4 provides a visual representation of the agreed-upon technology stack.

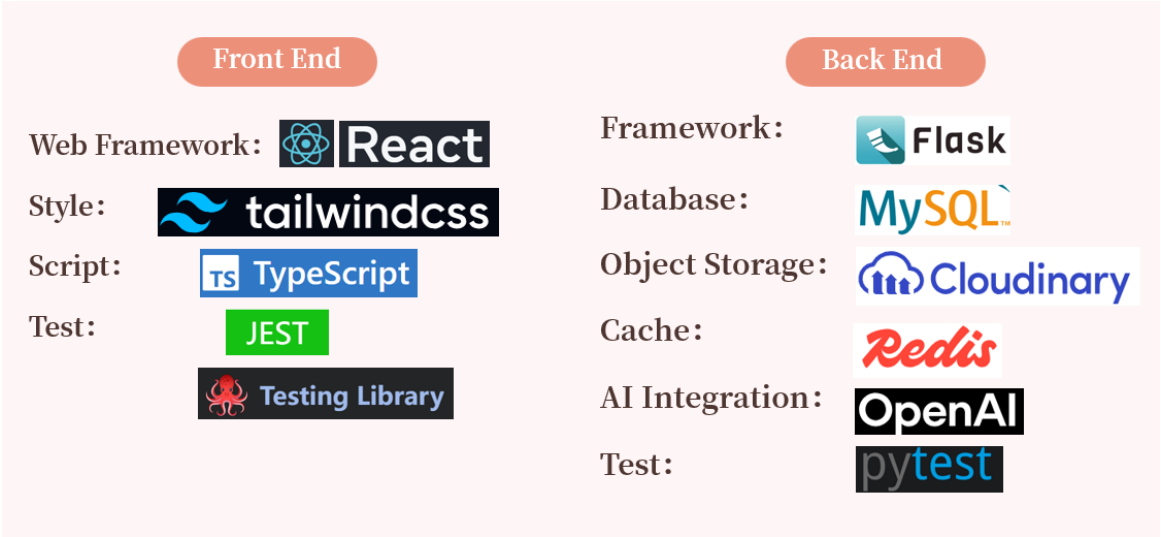

Figure 4: Database ER Diagram

3.1.1 Technologies for Frontend

- **Web Framework: React**
  We adopted **React** to build a dynamic, component-driven user interface. Its virtual DOM and efficient state management enable high-performance rendering and real-time UI updates. React's large ecosystem, including tools like React Router and Context API, supports modular and scalable development. This helps us create a responsive and interactive frontend experience tailored for our users.

- **Style: Tailwind CSS**
  Tailwind CSS is a utility-first CSS framework that allows us to rapidly develop consistent and mobile-friendly designs. Instead of writing custom CSS, we compose UI elements using utility classes, which speeds up development and ensures visual coherence. Tailwind also integrates seamlessly with React, supporting conditional styling through class bindings, which is essential for our design customization, such as adapting to the platform's pink color theme.

- **Scripting Language: TypeScript**
  TypeScript enhances JavaScript with static typing, enabling better code safety, editor autocompletion, and early error detection. In our project, it improves maintainability and developer collaboration by enforcing clear data structures and interfaces. This is particularly valuable as our codebase grows, ensuring fewer runtime errors and facilitating easier onboarding of new developers.

- **Testing: Jest & Testing Library**
  For frontend reliability, we employ **Jest** to write fast and isolated unit tests, and use **Testing Library** to simulate user interactions (such as clicks or form inputs) in a browser-like environment. This combination helps validate both the internal logic and the actual user experience, making sure our features work as intended across different states and edge cases.

3.1.2 Technologies for Backend

- **Framework: Flask**
  We selected **Flask**, a lightweight and flexible Python web framework, to build our RESTful backend APIs. Flask allows for rapid prototyping with minimal overhead while still supporting robust production deployment. Its modularity and clear routing system make it easy to structure our application, integrate with databases, and add extensions like authentication or request validation.

- **Database: MySQL**
  MySQL serves as our main relational database, used to manage structured data such as user profiles, blog posts, forum discussions, and job listings. Its support for ACID transactions ensures data consistency and integrity. We also utilize indexing and foreign key relationships to enhance query efficiency and maintain complex data relations.

- **Object Storage: Cloudinary**
  To manage multimedia assets, we use **Cloudinary**, which offers optimized storage, automatic image/video transformations, and delivery via a global content delivery network (CDN). This significantly reduces load time and bandwidth usage, especially critical for pages with many images like user avatars, blog covers, and job posters.

- **Caching: Redis**
  Redis is employed as an in-memory data store to cache frequently accessed data, such as trending posts and popular job listings. This reduces the number of database queries and dramatically improves performance under high traffic. We also use Redis for storing session tokens and temporary data, contributing to a smoother and faster user experience.

- **AI Integration: OpenAI**
  We integrate **OpenAI**'s APIs to provide intelligent, AI-powered features on our platform. This includes automatic content summarization, keyword extraction from user posts, and smart Q&A support. These features enhance accessibility and usability by helping users find relevant content faster and interact with the platform more effectively.

- **Testing: Pytest**
  On the backend, we use **Pytest** to write modular, scalable tests for our business logic and API endpoints. Pytest's fixtures and powerful assertion capabilities allow us to simulate different scenarios and edge cases. This ensures that new code changes do not introduce bugs and that our backend behaves consistently under various conditions.

3.2 Database Design

The database design is focused on delivering a simple and efficient solution for managing extensive amounts of data essential for supporting the detailed analytical capabilities of the program. Specifically, the database is structured into the following tables:

1. Users Table
2. Blog Table

3. Post Table
4. Topic Table
5. Comment Table
6. Post_Likes Table

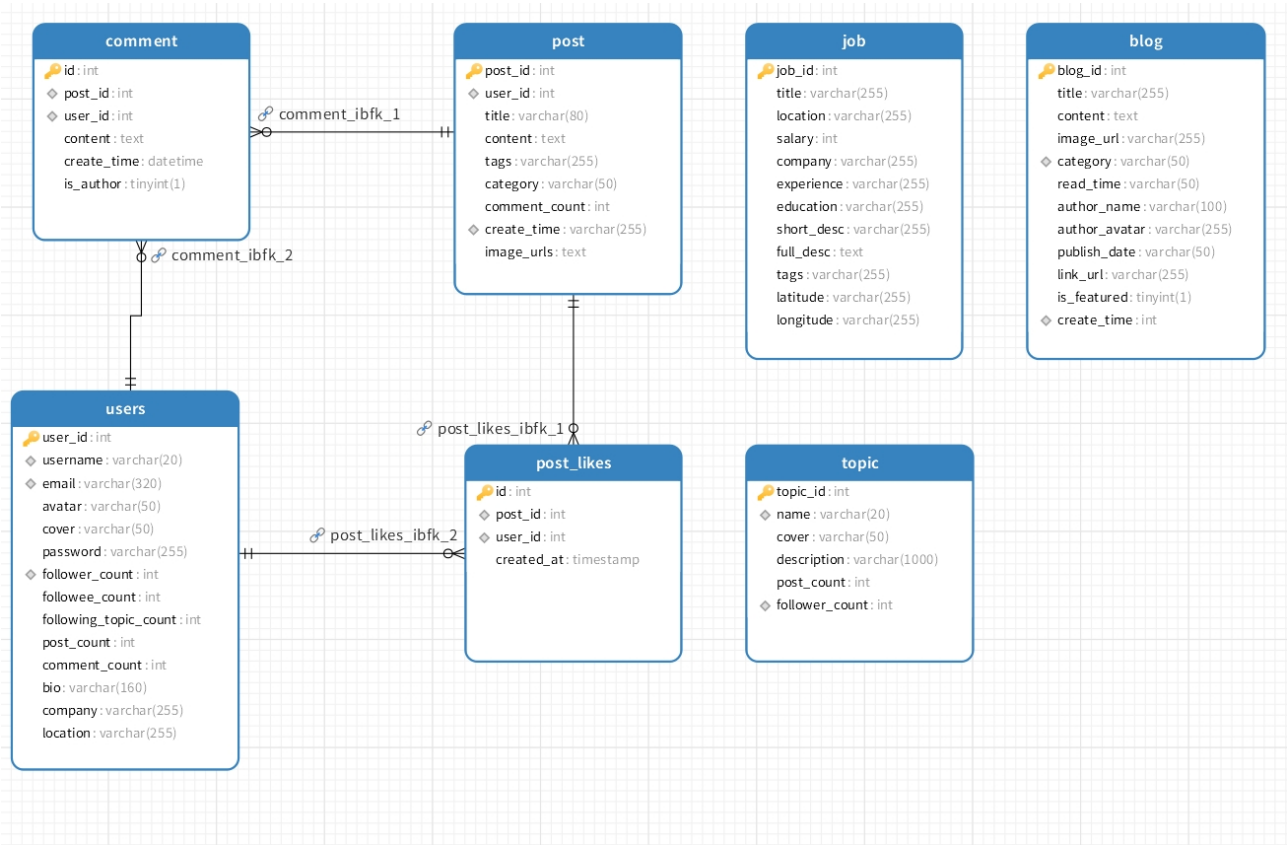

Figure 5: Database ER Diagram

This segmentation of tables enables organized storage and retrieval of data, ensuring that the system can effectively handle the diverse information required for the program's analytical functionalities. The roles and relationships of these tables are depicted in the Entity-Relationship (ER) diagram, as illustrated in Figure 5. The ER diagram offers a holistic view of how the tables are interconnected and how data moves between them. The specific functionalities and purposes of each table are explained in detail in the subsequent sections, following the order as mentioned above.

3.2.1 Users Table

The Users table stores information about registered users. Attributes like userId serve as unique identifiers, while username, email, avatar, cover, and bio capture personal details. Password stores user login credentials, followerCount tracks followers, followeeCount counts the number of users followed, followingTopicCount keeps track of topics followed, postCount records the number of posts created, commentCount tracks the number of comments made, company and location store additional user profile details.

3.2.2 Comment Table

The Comment table stores individual comments made by users on various content within the system. It includes attributes such as commentId for identification, commentableId to associate the comment with a specific entity, commentableType to specify the type of entity being commented on, userId to denote the user who made the comment, content for the actual text of the comment, and createTime to track when the comment was created.

3.2.3 Post Table

The Post table contains details about user-generated posts within the system. Attributes like postId serve as unique identifiers, userId links the post to the user who created it, title and content store the post's title and

main text, tags allow for categorization, category classifies the post into specific topics, commentCount tracks the number of comments on the post, createTime records the post creation time, and imageUrls hold links to any associated images with the post.

3.2.4 Topic Table

The Topic table represents different discussion topics or categories within the platform. It includes topicId for identification, name for the topic title, cover for displaying an image associated with the topic, description for a brief overview, postCount for tracking the number of posts under the topic, and followerCount indicating how many users follow the topic.

3.2.5 Blog Table

The Blog table manages blog-related content in the system. It includes attributes such as blogId for identification, title for the blog title, content for the main blog text, imageUrl for any related images, category for categorization, readTime to estimate reading duration, authorName and authorAvatar for author details, publishDate for publication time, linkUrl for accessing the blog externally, isFeatured to highlight featured blogs, and createTime to record creation time.

3.2.6 Job Table

The Job table manages employment opportunity data within the system. It includes attributes such as jobTitle for the name of the job, location for the job's city, and salary to indicate monthly compensation. jobId uniquely identifies each job entry, while companyName records the employer's name. experienceRequirement and educationRequirement capture the necessary qualifications, if any. The table also contains shortDescription for a brief summary and detailedDescription for an in-depth explanation of job duties and expectations. The tags field is used to categorize jobs (e.g., "Women-Friendly", "Entry Level"), while latitude and longitude store the geographical coordinates of the job's location, enabling map-based search or filtering.

3.2.7 Post_Likes Table

The Post_Likes table records user interactions with posts in the form of "likes." Each row represents a unique like action and is identified by the id field. The post_id attribute links the like to a specific post, while user_id indicates which user performed the action. The created_at field stores the timestamp of when the like occurred, providing a chronological view of engagement. This table enables features such as counting post popularity, tracking user activity, and supporting personalized content feeds based on user interests. It also facilitates analytics on post engagement trends over time.

4 Technical Implementation

4.1 Front-end

The front-end design of HerBlossom website aims to provide an engaging user experience, allowing women worldwide to easily share posts and interact through comments. With interactive interfaces and responsive layouts, the design focuses on accessibility and internationalization to ensure users can access the site effortlessly regardless of location or language, promoting cross-cultural interactions and learning. This design will help maintain the site's activity levels and create a friendly, diverse, and vibrant online community.

4.1.1 Technology Stack

Our front-end architecture at HerBlossom is centered around the Python Flask framework, a well-established micro web framework known for its lightweight nature and robust capabilities. This framework enables rapid development and seamless integration with our backend services. Leveraging our team's expertise in Python, we are able to fully use Flask's features to build a strong, sustainable codebase for our platform.

4.2 Back-end

The backend system of HerBlossom website plays a crucial role in managing and providing the data needed for the platform's various features, including content display and user interactions. Developed in alignment with the frontend requirements, the backend focuses on simplicity, scalability, and performance to support the smooth functioning of HerBlossom's diverse functionalities and user engagement.

4.2.1 Database Management

The back-end system of the HerBlossom website is designed to handle data management, user interaction and back-end security operations. It is built to complement the front end, ensuring that all functions are responsive, secure and scalable. This system adopts modern web technology and follows the best practices of software engineering to provide a reliable experience for users and administrators.

4.2.2 Account Authentication

HerBlossom utilizes a relational database (MySQL) to store and organize all platform data, including user profiles, blog posts, forum posts, and work lists. This mode is designed to ensure data integrity, support efficient queries, and support future scalability. Apply appropriate indexing and normalization techniques to enhance performance and maintain data consistency across different modules.

4.2.3 Input Validation

All user input, whether submitted through forms or apis, must undergo strict server-side validation to prevent malicious attacks such as SQL injection, XSS and CSRF. Verification includes data types, required fields, length constraints, and format verification (for example, valid email formats). Libraries and frameworks (such as Express-validator in Node.js) are used to simplify and execute consistent validation logic.

4.2.4 Location-Based Job Search

To support interactive map-based job discovery, each job posting in HerBlossom includes geographic coordinates (latitude and longitude). When a user selects a location on the map, the backend performs a geospatial query to retrieve job postings within a specified radius (e.g., 5 km). This feature uses spatial database extensions to enable fast, efficient location filtering. This capability allows users to find relevant opportunities nearby, enhancing the practical value of the job module.

4.3 Testing

To ensure the robustness and reliability of our project, we implemented a comprehensive and multi-layered testing strategy.

4.3.1 User Documentation Testing

Comprehensive user documentation was prepared and distributed to a designated group of users for thorough testing during Week 11. This approach enabled users to engage with the system holistically, offering valuable feedback on usability and identifying potential issues from an end-user perspective. Based on the recommendations from the testing team, we conducted detailed discussions and made targeted adjustments to the project. The insights obtained during this phase significantly contributed to enhancing the overall user experience and refining the system's functionality.

4.3.2 Unit Testing

Unit tests were integrated throughout the development lifecycle to verify the correctness of individual components. By ensuring that each part of the codebase behaves as expected, unit testing reduces the risk of bugs and logical errors. This proactive and systematic approach facilitates early detection and resolution of issues, thereby improving the overall stability, maintainability, and reliability of the system.

4.3.3 API Testing with Python Scripts

We automated API testing using Python scripts that send HTTP requests containing a fixed authentication token to validate the functionality of each endpoint. These scripts cover a wide range of request scenarios and verify that both the status codes and returned data meet expectations. A successful run—where all assertions pass—indicates that the relevant API functionality is performing as intended. This ensures system stability and supports seamless integration between components.

4.4 Deployment

Deployment is a crucial phase in web development. Proper setup of the project environment and resource delivery is essential for protecting customer data and ensuring the stable operation of production systems. To improve the efficiency of system updates and maintenance, we adopted an automated deployment strategy.

4.4.1 Continuous Integration with GitHub Actions

To streamline infrastructure deployment, we implemented continuous integration (CI) using GitHub Actions. Through automated deployment pipelines, the system is deployed to a remote server upon each commit to the repository. This automation enhances deployment efficiency, ensures consistency across environments, and supports real-time testing of changes. GitHub Actions plays a key role in accelerating development workflows while maintaining reliability and traceability.You can see the process sample in Figure6.

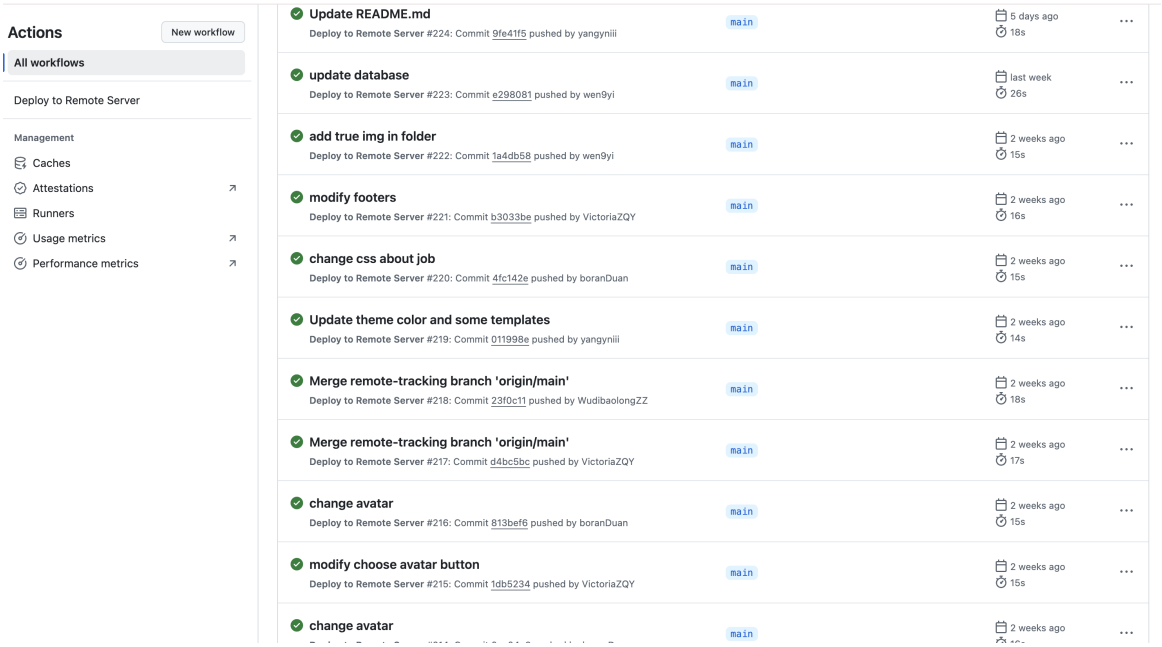

Figure 6: Continuous Integration Sample

4.4.2 System Optimization with Parcel

To ensure efficient resource management and fast load times in the production environment, we utilized **Parcel** as our web application bundler. Parcel is a zero-configuration web application bundler that offers lightning-fast performance, automatic dependency resolution, and built-in support for modern JavaScript features. To further enhance performance, we adopted **code splitting** strategies enabled by Parcel. This technique divides the application into smaller chunks that are loaded on demand, reducing the initial load time. In conjunction with **lazy loading**, only the necessary parts of the application are fetched and rendered when required, enhancing user experience on slower connections or lower-end devices. During development, Parcel's built-in **Hot Module Replacement** (HMR) allows developers to see changes in real-time without full-page reloads. This feature significantly boosts development productivity by providing immediate feedback and reducing the time needed to test and iterate on UI components.

4.5 Maintenance

The maintenance phase is vital for preserving a system's robustness, efficiency, and adaptability. Establishing standardized maintenance protocols ensures consistent practices, minimizes technical debt, and promotes long-term system sustainability. ~~This section outlines key tasks to be regularly undertaken to ensure long-term system performance and responsiveness to evolving user needs and technological advancements.~~

4.5.1 Performance Monitoring

System performance is regularly evaluated through key performance indicators and user feedback. A routine monitoring schedule (e.g., weekly or monthly checks) should be enforced to detect anomalies early. Ongoing monitoring enables prompt identification and resolution of issues, supporting system stability and responsiveness. All performance incidents and resolutions should be logged to support historical analysis and root cause tracing. ~~In addition, structured analysis of user feedback provides valuable insights into user needs, guiding future improvements and feature development.~~

4.5.2 API Updates

Staying informed about changes in the Google Programmable Search Engine API is essential. In a continuously evolving technological environment, APIs are frequently updated to enhance features and efficiency. A version-tracking mechanism should be in place, and designated personnel must be responsible for monitoring upstream API changelogs. By promptly adapting to these updates, we maintain compatibility, gain access to new functionalities, and ensure the stability of the system—thus contributing to an improved user experience.

4.5.3 Data Source Review

Regular reviews and updates of data sources are conducted to ensure relevance and accuracy. A quarterly audit of all external data sources should be documented, noting changes, deprecations, or reliability concerns. ~~As external data, such as policy-related content, may change over time, systematic updates help maintain the reliability of the information used by the system.~~

4.5.4 Documentation and Comments

We have developed detailed and well-organized documentation, along with comprehensive inline comments, to support future maintenance and facilitate collaboration. All documentation should follow a consistent style guide (e.g., Markdown with specific headers for APIs, modules, and examples). The documentation covers system architecture, core features, and API specifications. Any major code updates must be accompanied by corresponding documentation revisions to prevent inconsistencies. This ensures that future developers can quickly understand, troubleshoot, and extend the system as needed.

5 Conclusion

The development of HerBlossom has been a valuable project that combines technical implementation with a meaningful social mission — to empower women through knowledge sharing, job access, and community support. The platform integrates multiple modules including a blog, forum, job recommendation system, and an interactive map, all supported by a robust backend and a responsive frontend. This section summarizes future directions, team reflections, and acknowledgments.

5.1 Future Improvements

In future versions, we may add the following contents, such as:

- ~~Project progress network diagram~~
- ~~Non-functional requirements description~~
- ~~Use case diagram~~
- ~~The sequence diagram of the core use cases~~
- ~~Enhance interactivity with users~~
- ~~The application will include multilingual support to meet the needs of a diverse and inclusive user base.~~
- ~~Add ai functions to recommend relevant content to users through intelligent algorithms.~~
- ~~Our goal is to enhance the security and privacy features of the system to ensure that user data is protected.~~

5.2 Team Reflections

The development process of HerBlossom has provided the team with hands-on experience in full-stack web development, project coordination, and user-centered design. Key takeaways include:

- **Collaborative Development:** Close coordination between frontend and backend teams was essential to ensure consistent data flow and smooth user experience.
- **Design Thinking:** Emphasizing the needs of women users helped guide UI/UX decisions and content structure.
- **Technical Growth:** Team members gained deeper knowledge in technologies such as RESTful API design, database normalization, map API integration, and authentication mechanisms.
- **Problem Solving:** Challenges such as data synchronization, input validation, and role-based access control encouraged practical problem-solving and teamwork.

5.3 Acknowledgments

We sincerely thank the following individuals and groups:

- **Professor:** She inspired our idea for this project and provided us with continuous feedback during the development process.
- **HeadTA and our TA**: They provide guidance, feedback and support throughout the entire project life cycle. Share her previous development experience with us.
- **Women's Federation (Client Organization) :** Share valuable insights into the actual needs of users and help us focus on meaningful impacts.
- **Open source community:** Its tools and libraries are an indispensable part of building a platform.
- **Test users:** Friends, classmates and volunteers who participate in the usability test and provide constructive feedback.

HerBlossom reflects not only technical achievement but also a commitment to social good. We hope the platform continues to evolve and support women in diverse areas of life and career.

References

[1] GitHub Actions. GitHub Actions Documentation. GitHub. https://docs.github.com/en/actions

[2] Google Programmable Search Engine API. Programmable Search Engine API Developer Guide. Google. https://programmablesearchengine.google.com/

[3] Leaflet. Leaflet: An Open-Source JavaScript Library for Mobile-Friendly Interactive Maps. https://leafletjs.com/

[4] OpenAI. OpenAI: AI Research and Deployment Company. https://www.openai.com/