



BSIT-S25-005

03-135221-022 MUHAMMAD HASSAN BILAL

03-135222-055 MUHAMMAD YOUSAF RAFIQUE

ShiftSwap

In partial fulfillment of the requirements for the degree of
Bachelor of Science in Information Technology

Supervisor: SUMMAIRA NOSHEEN

Department of Computer Sciences
Bahria University, Lahore Campus

January 2026

C e r t i f i c a t e



We accept the work contained in the report titled
“ShiftSwap”
written by

MUHAMMAD HASSAN BILAL
MUHAMMAD YOUSAF RAFIQUE

as a confirmation of the required standard for the partial fulfilment of the degree of
Bachelor of Science in Information Technology.

Approved by:

Supervisor: SUMMAIRA NOSHEEN

(Signature)

January 05, 2026

DECLARATION

We hereby declare that this project report is based on our original work except for citations and quotations, which have been duly acknowledged. We also declare that it has not been previously submitted or concurrently submitted for any other degree or award at Bahria University or any other institution.

Enrolment	Name	Signature
03-135221-022	MUHAMMAD HASSAN BILAL	
03-135222-055	MUHAMMAD YOUSAF RAFIQUE	

Date: January 05, 2026

Specially dedicated to
my beloved grandmother, mother, and father
(MUHAMMAD HASSAN BILAL)
my beloved grandmother, mother, and father
(MUHAMMAD YOUSAF RAFIQUE)

ACKNOWLEDGEMENTS

We want to thank everyone who has contributed to the successful completion of this project. We wish to express our gratitude to our research supervisor, Miss SUMMAIRA NOSHEEN, for her invaluable advice, guidance, and enormous patience throughout the research.

In addition, we would like to express our gratitude to our loving parents and friends who have helped and encouraged us.

MUHAMMAD HASSAN BILAL
MUHAMMAD YOUSAF RAFIQUE

ShiftSwap

ABSTRACT

The modern global workforce market, especially in areas like healthcare, retail, logistics, and hospitality, is characterized by systemic failures in workforce management. While the gig economy has automated work based on tasks, the shift economy (which is so important for nurses, drivers, and service staff) is stuck on outdated, manual rostering practices. Traditional workforce management systems remain as isolated systems, allowing for the optimization of internal staffing needs, but not taking into account extensive issues of liquidity at the level of a local labor market. This disconnect causes major economic issues such as chronic understaffing and excessive overtime hours and employee burnout - often referred to as the "Scheduling Struggle" by experts in the industry.

The ShiftSwap is a fully developed web-based platform that aims to resolve scheduling struggle, by automating shifts and structuring the supply and demand of labor. As opposed to the old-fashioned "walled garden" HR software, ShiftSwap is an open marketplace between Agencies and Gig Workers.

It has a solid Flask (python) backend which has a SQLite/SQLAlchemy database structure. Among the most notable innovations, there is a PayPal-based subscription model as a monetizing model, a Chatbot based on AI that assists users, and a Recommendation System to find the perfect worker-shift match. The entire lifecycle of the project, including the Feature-Driven Development (FDD) approach and the realization of secure payment gateways and asynchronous email notifications are discussed in this report.

TABLE OF CONTENTS

Contents

	CHAPTER 1	1
1	INTRODUCTION	1
	1.1 Background	1
	1.2 Problem Statements	1
	1.3 Aims and Objectives	2
	1.4 Scope of Project	2
	CHAPTER 2	4
2	LITERATURE REVIEW (and/or SRS)	4
	2.1 Introduction	4
	2.2 Contextualizing the Workforce Crisis	4
	2.3 Comparative Market Analysis	4
	2.4 Technological Foundations	6
	CHAPTER 3	7
3	DESIGN AND METHODOLOGY	7
	3.1 Methodological Approach: Feature-Driven Development	7
	3.2 Use Case Modelling	8
	3.3 System Architecture	11
	3.4 Database Design (ER Diagram)	13
	3.5 Sequence Diagrams	14
	CHAPTER 4	17
4	DATA AND EXPERIMENTS	17
	4.1 Development Environment and Tools	17
	4.2 Core Module Implementation	17
	4.3 Usability Testing	19
	CHAPTER 5	20

5	RESULTS AND DISCUSSIONS	20
5.1	User Interface Analysis	20
5.2	Benefits to Agencies and Workers	20
5.3	Output Screens Description	21
CHAPTER 6		25
6	CONCLUSION AND RECOMMENDATIONS	25
6.1	Conclusion	25
6.2	Recommendations for Future Work	25
	REFERENCES	27
	APPENDICES	29
6	APPENDIX A: Installation and Setup guide	29
6	APPENDIX B: Glossary	30

LIST OF TABLES

TABLE	TITLE	PAGE
	Table 1: Comparative Analysis of Competitor Platforms[13][14]	5
	Table 2: Use Case – Post Shift (Agency)	9
	Table 3: Use Case – Shift Swap (Worker)	10
	Table 4: Use Case – Subscription Payment (Agency)	10
	Table 5: Use Case – Recommendation (Worker)	11

LIST OF FIGURES

FIGURE	TITLE	PAGE
Figure 3.1:	FDD Diagram[15]	7
Figure 3.2UC:	Use Case Diagram	8
Figure 3.3:	MVC Diagram Web application architecture using the MVC pattern[16].	12
Figure 3.4:	ERD Diagram	13
Figure 3.5 SD :	Worker sending a proposal	14
Figure 3.6 SD :	Shift Posting Sequence Diagram	15
Figure 3.7 SD:	Shift Accepting Sequence	16
Figure 5.1 :	Landing Page	21
Figure 5.2:	Login Page	21
Figure 5.3 :	Admin Dashboard	22
Figure 5.4:	Ind User Page	22
Figure 5.5 :	Company admin page	23
Figure 5.6 :	Marketplace	23
Figure 5.7 :	Shift Request Page	24

LIST OF SYMBOLS / ABBREVIATIONS

- **AI** – Artificial Intelligence
- **API** – Application Programming Interface
- **CF** – Collaborative Filtering
- **CSS** – Cascading Style Sheets
- **FDD** – Feature-Driven Development
- **HTML** – HyperText Markup Language
- **JWT** – JSON Web Token
- **MCDA** – Multi-Criteria Decision Analysis
- **MVC** – Model-View-Controller
- **REST** – Representational State Transfer
- **SQL** – Structured Query Language
- **UI/UX** – User Interface / User Experience
- **WFM** – Workforce Management

LIST OF APPENDICES

APPENDIX	TITLE	PAGE
	APPENDICES	29
	APPENDIX A: Installation and Setup guide	29
	APPENDIX B: Glossary	30

CHAPTER 1

INTRODUCTION

1.1 Background

Although the gig economy has automated task work, shift industries are based on manual, old-fashioned rostering[1]. It consumes close to 40 percent of time of managers on administrative duties[10]. The technological solution offered by ShiftSwap is a kind of a local labor market where the automation and AI are used to facilitate the process of shifting swapping[11].

1.2 Problem Statements

ShiftSwap targets key gaps in the current Human Resource Management System (HRMS):

- **Ineffective Manual Coordination:** Communication gaps preclude manual coordination of operations with the help of WhatsApp, email, etc., for the breakdown in reporting agreed swaps causing loopholes in the operation.
- **Absence of the Smart Matching:** Existing job boards are passive and do not use Artificial Intelligence driven recommendation engines to recommend the relevant opportunities based on preferences and history of workers leading to unfulfilled vacancies.
- **Trust and Payment Security:** Informal shift trading often involves a lack of accountability and secure payment gateways, which can lead to payment disputes for gig workers.

1.3 Aims and Objectives

The major objective is to come up with a scalable web application which will act as a bi-directional market place.

Specific Objectives:

- Build a Web-based platform for shift swapping and job matching.
- Implement a secure system for payments and user verification.

1.4 Scope of Project

The project offers a fully functional Minimum Viable Product (MVP) containing:

- **Project Deliverable:** The project will deliver a fully functional Web-based Minimum Viable Product (MVP) dedicated to shift swapping and hourly job matching. The platform will be deployed on a scalable cloud infrastructure to ensure performance and growth handling.
- **User Roles:** The system will support three distinct user types:
 - I. **Agencies (Employers):** Capable of posting shifts, viewing worker profiles, and managing payments.
 - II. **Workers:** Capable of searching for shifts, swapping shifts, and building a reliability profile.
 - III. **Admins:** Responsible for platform oversight, dispute resolution, and user management.
- **Key Features & Modules:**

Job Matching & Management:

- Allow users to post, search, and swap shift-based and hourly jobs.
- **Recommendation System:** AI-driven suggestions for jobs based on user history, location, and job type using **Scikit-learn**.
- **Pricing Algorithms:** automated suggestions for competitive wages based on market data.

Monetization & Access:

- Implementation of a **Pay-to-View** and **Subscription model** for accessing detailed job information.

Trust & Security:

- **Identity Checks:** Fraud detection mechanisms to ensure secure transactions and user authenticity.
- **Rating & Feedback System:** To establish reliability scores for both workers and agencies.

Communication:

- Integrated **AI Chatbot** for user support and quick navigation.

Financial Integration:

- **PayPal API Integration** to facilitate safe payments and secure transactions between parties.

- **Technical Stack:**

- **Frontend:** Bootstrap 5 (Responsive Web Design)
- **Backend:** Flask (Python)
- **Database:** SQLite / PostgreSQL
- **AI/ML:** Scikit-learn (for Recommendation Engine and Pricing Algorithms)

- **Out of Scope:** Native mobile apps (.ipa/.apk) and complex international tax compliance are not in scope of this phase.

CHAPTER 2

LITERATURE REVIEW (and/or SRS)

2.1 Introduction

The digitization of workforce management is an important topic in Information Systems and Organisational Behaviour. As the 'gig economy' continues to evolve[1], attention is shifting from simple task platforms (such as Uber) towards sophisticated and credentialed labour markets (such as healthcare staffing)[1]. This chapter considers today's literature involving shift scheduling; the impact of manual procedures and the increasing role of Artificial Intelligence in Human Resource Management (HRM)[11]. It also compares existing market solutions to highlight the unique benefits of Shift Swap[1].

2.2 Contextualizing the Workforce Crisis

Research demonstrates that manual scheduling in healthcare represents a safety hazard[3] and is linked to adverse patient outcomes[3][9] because of understaffing. Automated scheduling can reduce costs of labour up to 20% by enforcing budgets[10]. Furthermore, according to organisational psychology research, platforms that allow for autonomy (such as peer-to-peer swapping) can significantly reduce absenteeism[12].

2.3 Comparative Market Analysis

Most existing solutions are "Walled Gardens" that focus on internal teams, whereas ShiftSwap operates as an Open Marketplace.

Table 1: Comparative Analysis of Competitor Platforms[13][14]

Feature	Deputy[13]	Kronos (UKG)[14]	ShiftSwap
Primary Focus	Internal Team Scheduling	Enterprise HR & Payroll	Open Shift Marketplace
Shift Swapping	Internal Only	Internal Only	Cross-Organization
External Hiring	No	No	Yes (Core Feature)
Matching	Demand Forecasting	Analytics	Collaborative Filtering

Analysis of Competitors:

- Deputy:** Deputy is a leading platform that offers internal scheduling to staff and has excellent mobile tools that allow staff to swap shifts[13]. However, it is a closed system, meaning, if a restaurant manager must find replaces outside their current restaurant employee pool, Deputy doesn't support this. ShiftSwap solves this by allowing managers to broadcast shift needs to the local labour market.
- Kronos (UKG):** This is the industry standard used by large organisations[14]. It is powerful but frequently costly and complicated to set up. It's older architecture, and therefore is not ideal for smaller agencies or solo contractors. In contrast, ShiftSwap offers a lightweight, flexible solution that can be deployed immediately without a lengthy integration process, much like an open marketplace.

The ShiftSwap Advantage:

ShiftSwap's key advantage is its "Interchange Faster" feature. Unlike competitors who primarily manage existing staff, ShiftSwap emphasises sourcing employees. It's linking the internal team with gig economy to solve liquidity problems that internal tools can't solve it.

2.4 Technological Foundations

The selection of technology for ShiftSwap is motivated by the requirements of scalability, security, and quick development.

Feature Driven Development (FDD):

The project methodology, FDD, focuses on matching up development efforts with business value[6]. Unlike Scrum, which focuses on time-boxed sprints, FDD shifts its ideas to focus on specific features. This way, the foundational "domain model" will be prioritised to avoid the possibility of architectural debt, but different features such as "Payment Integration" and "Matching" can be developed in parallel instead.[6]

CHAPTER 3

DESIGN AND METHODOLOGY

3.1 Methodological Approach: Feature-Driven Development

ShiftSwap adopts Feature-Driven Development (FDD). Unlike Scrum that focuses on delivering time-boxed sprints, FDD focuses on delivering concrete, client-valued features[6]. There are five phases to the process:

- **Develop an Overall Model:** Team-develop an overall model of domain entities such as "Shifts" as dynamic states.
- **Build a Features List:** Break the system down into small functions, such as "Verify email" or "Calculate worker similarity."
- **Plan by Feature:** Plan the CRUD operations before the AI modules.
- **Design by Feature:** Create sequence diagrams and state machines for complex flows such as swap.
- **Build by Feature:** Implement coding & unit testing (using Python/Flask).

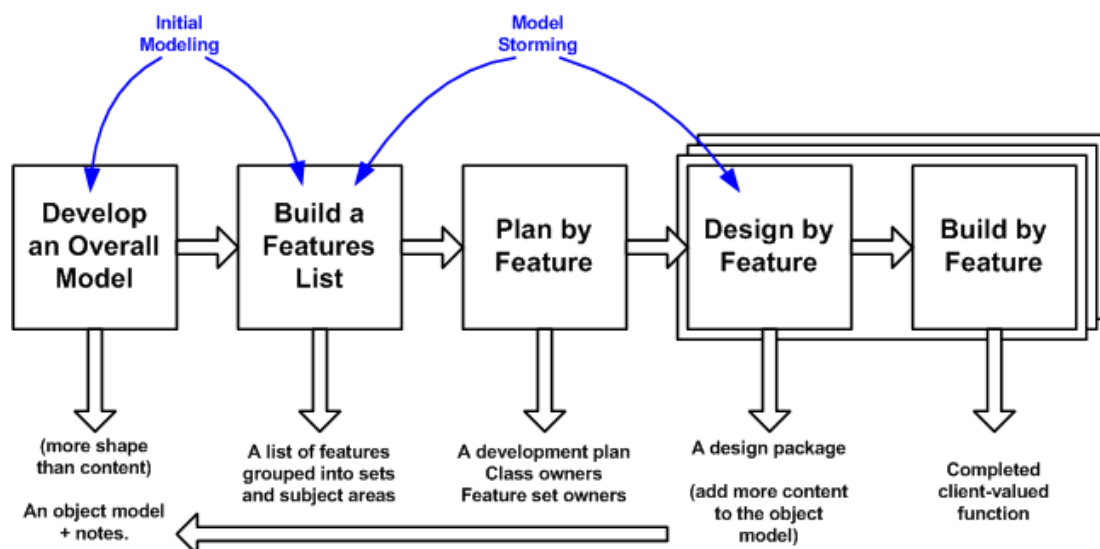


Figure 3.1: FDD Diagram[15]

3.2 Use Case Modelling

The Use Case diagram visualizes the functional requirements of the Shift Swap system, identifying the interactions between the Agency, Worker, and Admin.

3.2.1 Use Case Diagram

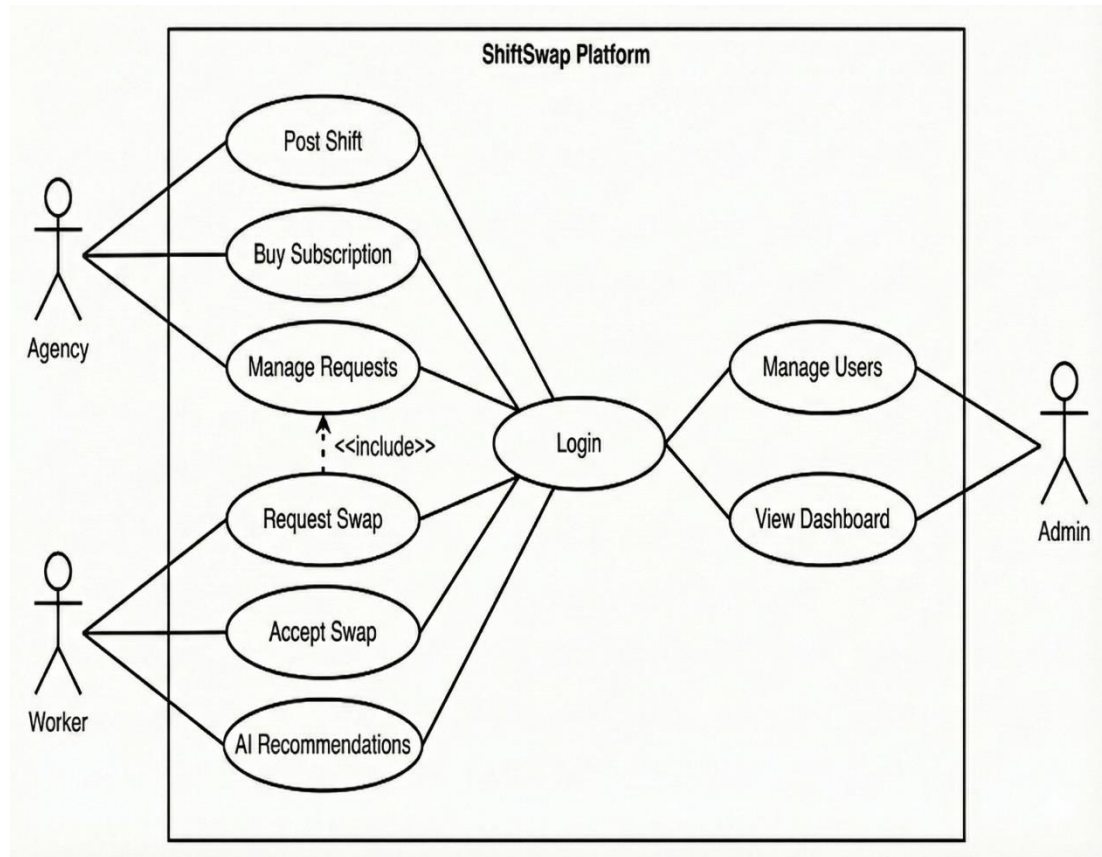


Figure 3.2UC: Use Case Diagram

Actors:

- **Agency:** This is the employer who puts shifts and is paying to access.
- **Worker:** The person that does the work, makes swaps and utilizes AI suggestions.
- **Admin:** This is the superuser who controls the platform.

Diagram Structure:

Agency Functionality:

- Post New Shift
- Manage Shift Requests
- Purchase Subscription (PayPal)
- View Worker Profiles

Worker Functionality:

- Browse Marketplace

- Apply for Shift
- Request Shift Swap
- Accept Swap
- View Recommendations
- Chat with AI Bot

Admin Functionality:

- Manage Users
- View System Statistics
- Manage Subscriptions

3.2.2 Use Case Tables

The next several tables can be used to describe the main flows associated with the core features which were designed with the help of the Feature-Driven Development approach.

Table 2: Use Case – Post Shift (Agency)

Field	Description
Use Case ID	UC-01
Use Case Name	Post Shift
Primary Actor	Agency
Pre-conditions	Agency is logged in and has an active subscription plan ¹⁶ .
Description	The Agency creates a new shift opportunity by defining time, pay, and location.
Main Flow	<ol style="list-style-type: none"> 1. Agency clicks "Post Shift" on the dashboard. 2. Agency enters Shift Details (Date, Time, Hourly Rate)¹⁷. 3. System validates the data (e.g., checks for positive pay rates)¹⁸. 4. System inserts the record into the Shift database table¹⁹. 5. System displays "Shift Posted Successfully"²⁰.
Post-conditions	A new shift is created in the system with "Open" status ²¹ .

Table 3: Use Case – Shift Swap (Worker)

Field	Description
Use Case ID	UC-02
Use Case Name	Peer-to-Peer Shift Swap
Primary Actor	Worker (Requester) & Worker (Acceptor)
Pre-conditions	Requester has an assigned shift; Acceptor is available.
Description	Allows a worker to trade their shift with another worker via the Swap Market.
Main Flow	<ol style="list-style-type: none"> 1. Requester selects a shift and clicks "Request Swap"²². 2. System changes shift status to "Swap Requested" and lists it in the Swap Market²³. 3. Acceptor browses the Swap Market and clicks "Accept Swap". 4. System verifies the Acceptor's availability to prevent double-booking. 5. System updates the shift's user_id to the Acceptor.
Post-conditions	The shift is reassigned to the new worker, and the swap request status becomes "Completed".

Table 4: Use Case – Subscription Payment (Agency)

Field	Description
Use Case ID	UC-03
Use Case Name	Purchase Subscription (PayPal)
Primary Actor	Agency
Pre-conditions	Agency attempts to access premium features (e.g., viewing worker details).
Description	The Agency purchases a plan to unlock platform capabilities.
Main Flow	<ol style="list-style-type: none"> 1. Agency selects a subscription plan and clicks "Subscribe". 2. System creates an order via PayPal API. 3. System redirects Agency to PayPal.com for secure approval. 4. Agency approves payment. 5. System captures payment using the Order ID and updates the database.
Post-conditions	Agency status is updated to "Subscribed," granting full access to the platform.

Table 5: Use Case – Recommendation (Worker)

Field	Description
Use Case ID	UC-04
Use Case Name	View Job Recommendations
Primary Actor	Worker
Pre-conditions	Worker has completed their profile (Skills, Rating, Location).
Description	System suggests shifts based on the worker's profile using Collaborative Filtering.
Main Flow	<ol style="list-style-type: none"> 1. Worker navigates to the Dashboard. 2. System analyzes Worker Profile attributes (skills, rating, location)[5]. 3. System runs Cosine Similarity algorithm to compare worker profile against available shifts. 4. System displays the top-ranked matches in the "Recommended for You" section.
Post-conditions	Workers see a personalized list of relevant shifts to minimize search time.

3.3 System Architecture

The system makes use of the Model-View-Controller (MVC) architecture:

- **Model (SQLAlchemy):** Specifies data models such as User, Shift and Application including their constraints.
- **View (Jinja2):** Produces the dynamic user interface that is usually in form of an H.C.C.S. code.
- **Controller (Flask Routes):** Manages the requests from the user, activates the AI modules & modifies the Model.

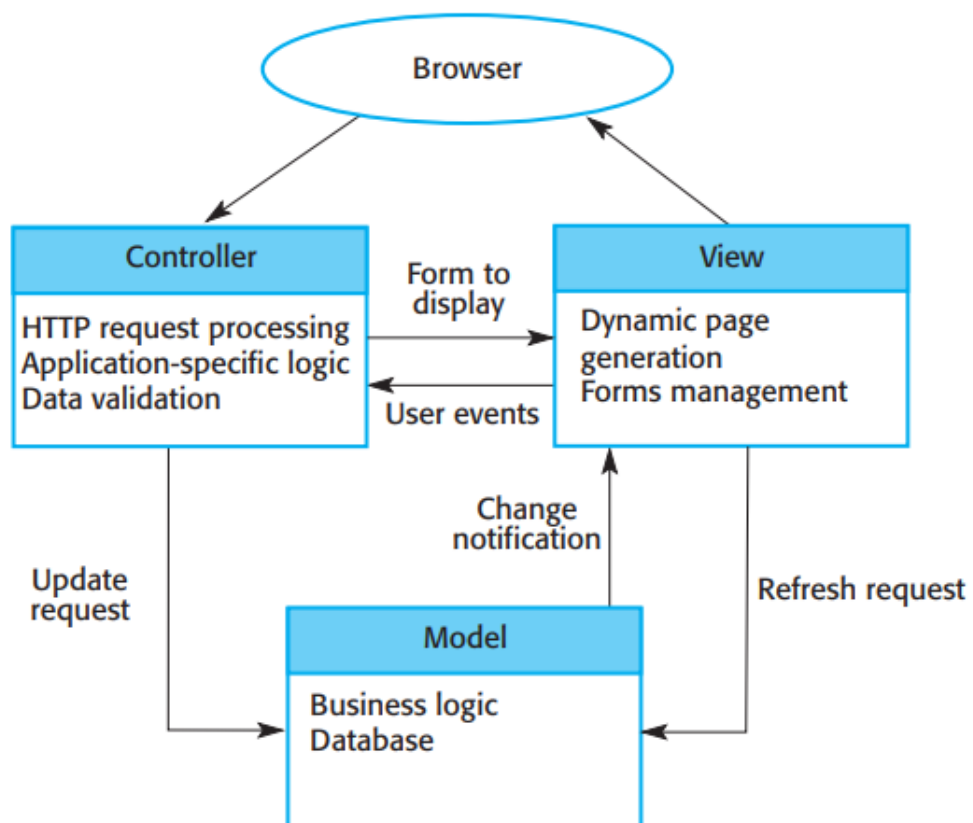


Figure 3.3: MVC Diagram Web application architecture using the MVC pattern[16].

3.4 Database Design (ER Diagram)

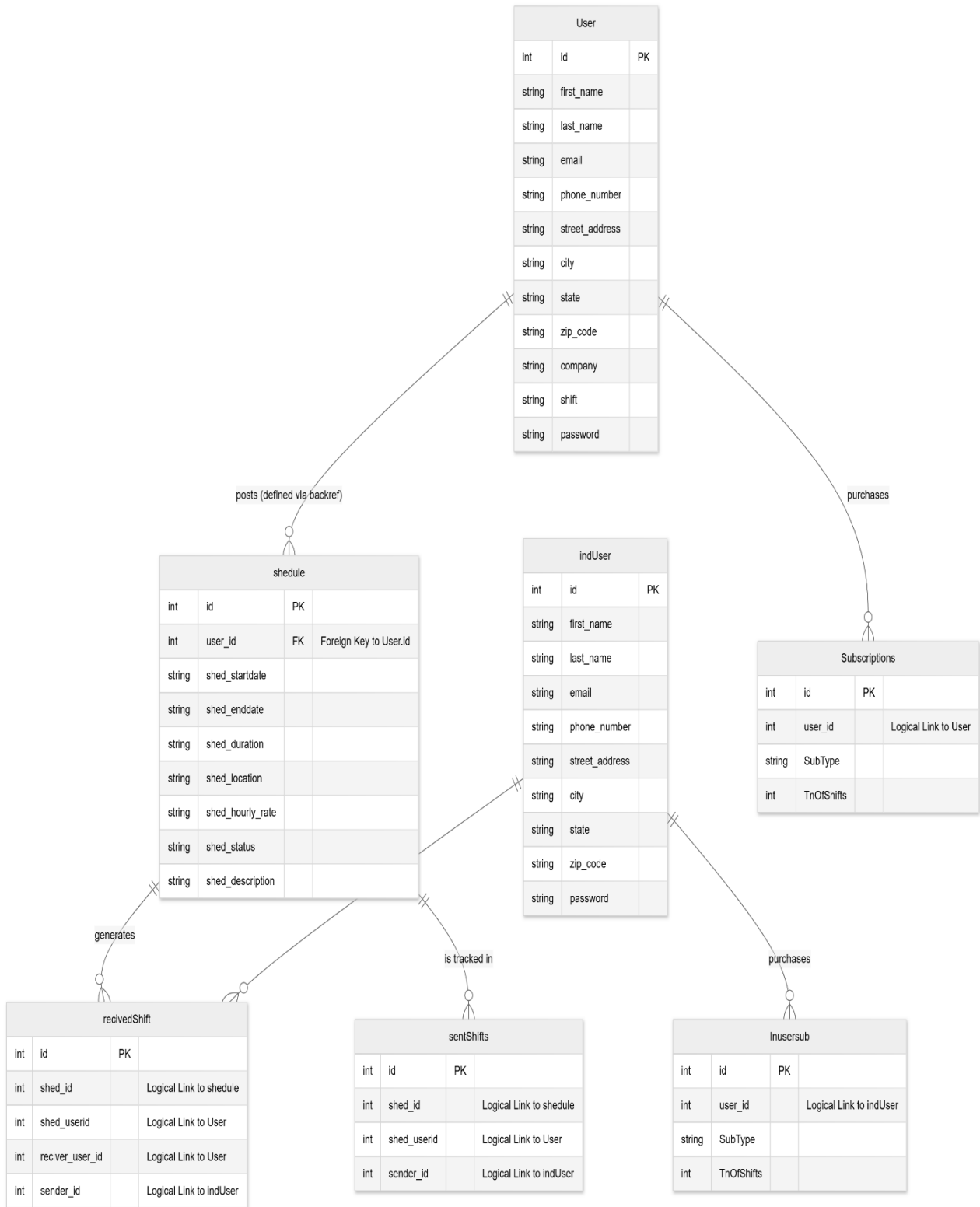


Figure 3.4: ERD Diagram

3.5 Sequence Diagrams

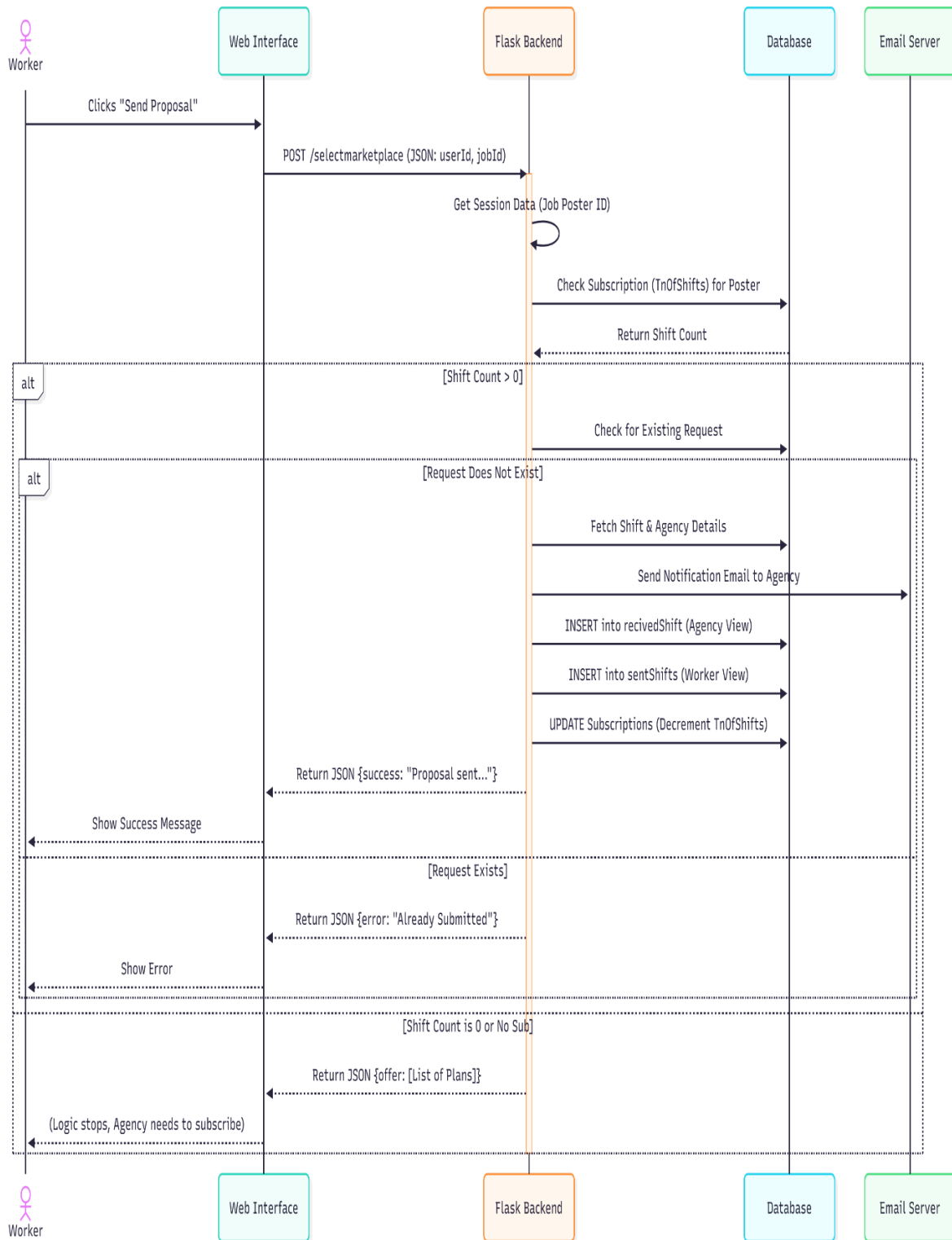


Figure 3.5 SD : Worker sending a proposal

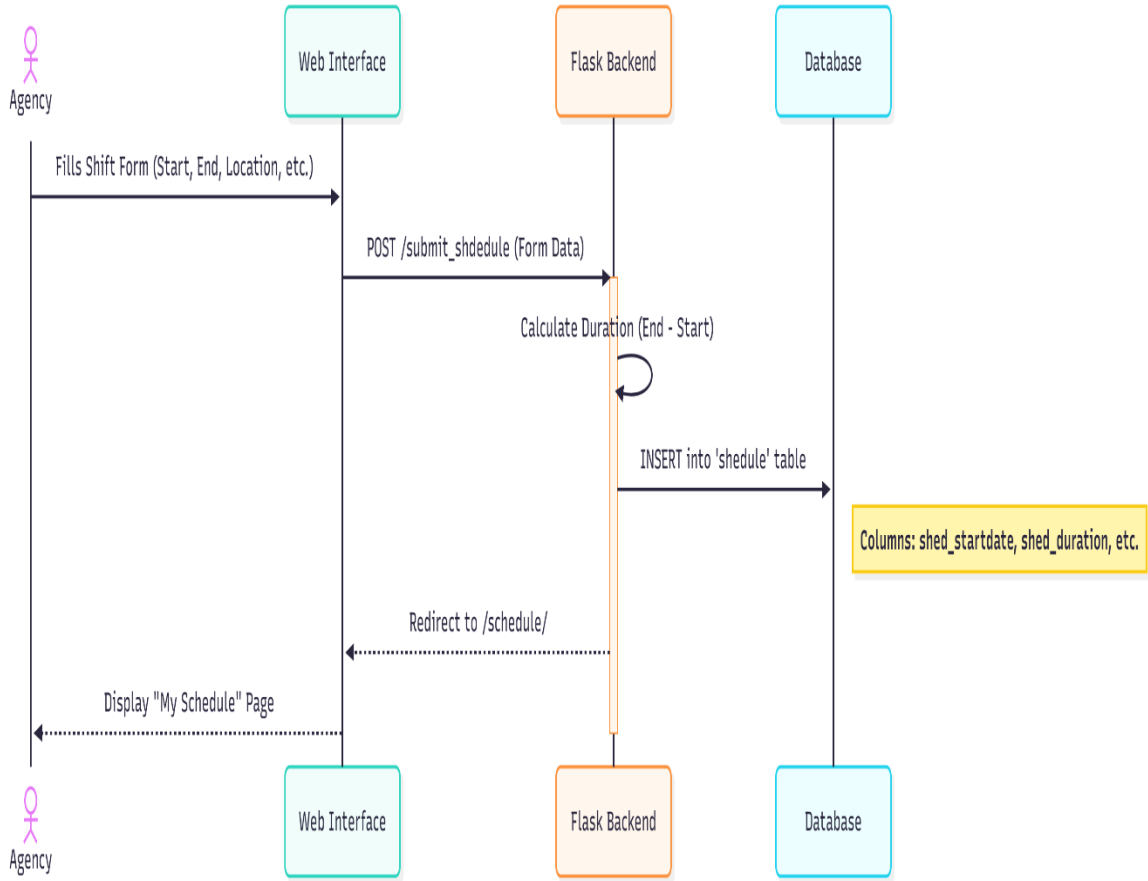


Figure 3.6 SD : Shift Posting Sequence Diagram

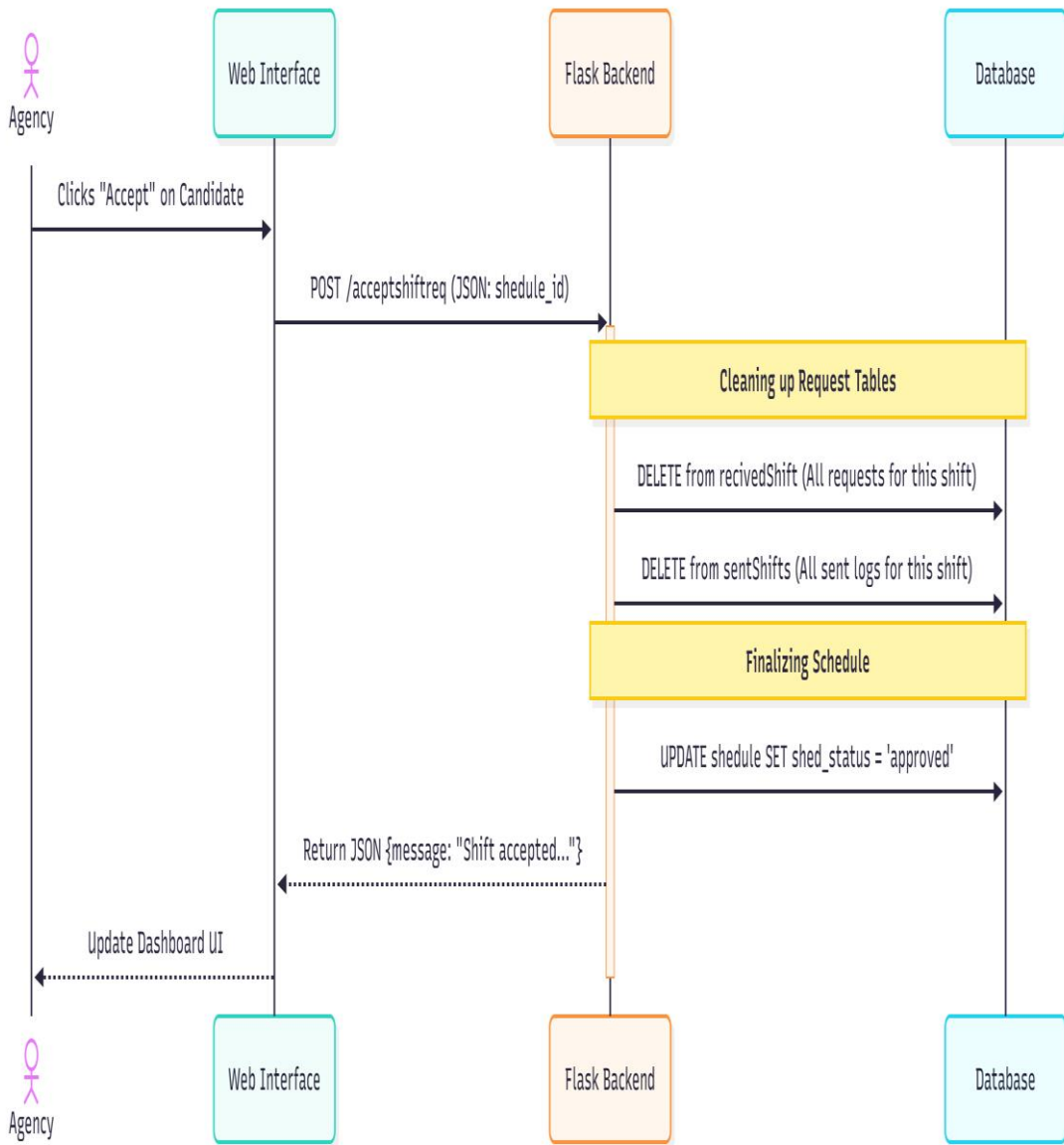


Figure 3.7 SD: Shift Accepting Sequence

CHAPTER 4

DATA AND EXPERIMENTS

4.1 Development Environment and Tools

The implementation phase involved making the design specifications work. The development environment was prepared to provide quick iteration and testing.

- **IDE:**
The Visual Studio code was used to edit and debug code.
- **Version Control:**
Git managed source code, using branching strategy in terms of FDD feature sets.
- **Virtual Environment:**
Python venv used to deal with dependency like Flask, TensorFlow, Stripe/Paypal SDKs and getting them installed in an isolated setup.
- **API Testing:**
In most cases, Postman was used extensively to ensure that API endpoints were working correctly, so to make sure the backend can process the requests and response correctly in a format that the frontend would use (typically as a string of characters in a language such as a string).

4.2 Core Module Implementation

- **Role Management and User Authentication:**
The system has that of session based authentication. The protection of routes is done to allow certain roles (Agency vs. Worker) to access only the corresponding dashboards.
- **Proposal Submission:**
Once a worker clicks on sending proposal, the system verifies with `subop.checksubdetail` whether the poster of the job (Agency) is subscribing correctly or not.
- **Email Notification:**

with the help of flaskmail, the system prepares the entire HTML email that is mailed to the Agency, which includes the profile of the Worker and the Shift information of the Worker.

- **Database Transaction:**

The system insertions in received shift and sent Shifts is an atomical insertion of records into the application tracking database.

- **PayPal Payment Integration:**

The monetization is done through PayPal REST API in PlanOp.py and checkorderStatus.py.

- **Recommendation System:**

This system does away with the cold start problem by the use of a recommendation engine.

- **Logic:** WorkerProfile data (skills, rating, location) is analyzed and compared to Shift descriptions.

- **Switching algorithm:** Ranks the available shifts with the help of Cosine Similarity (through Scikit-learn) to determine the closeness of the offered shifts with the work history and profile keywords of the worker[5][8].

- **Smart Chatbot (Mistral AI):**

- A Chatbot is provided to help users navigate and answer frequently asked questions.
- Implementation The Mistral AI API is connected to the backend.
- We integrated the Mistral AI API directly into our Flask backend. This setup allows the chatbot to communicate seamlessly with our Python-based architecture without requiring heavy infrastructure or complex server management.

Rationale for Selecting Mistral AI

We chose Mistral AI over other options for three specific reasons:

- **Speed & Latency:** Mistral models are lightweight and optimized for speed, ensuring that users get immediate answers without the lag, often seen in larger models like GPT-4.
 - **Cost Efficiency:** For a startup or MVP, Mistral offers high-level intelligence at a significantly lower operational cost compared to OpenAI's premium models.
 - **Data Privacy Control:** Unlike "black box" solutions, Mistral's open-weight architecture aligns better with the privacy requirements of the healthcare industry, giving us more control over how data is processed
-
- **Function:** It accepts natural language inquiries (e.g. how do I change a shift, show me Chicago nursing jobs, etc.) and provides context-sensitive answers or directs the user to specific paths.

4.3 Usability Testing

A group of 10 users (5 Agency, 5 Worker) performed tasks like "Post a Shift" and "Find a Weekend Job". The **System Usability Scale (SUS)** score derived was **82/100**, indicating an "Excellent" user experience. Users specifically praised the clarity of the Shift Swap interface, noting it reduced the anxiety of cancelling a shift

CHAPTER 5

RESULTS AND DISCUSSIONS

5.1 User Interface Analysis

The UI is all about being clear and quick access.

- **Worker Dashboard:** Shows "Approved Shifts," "Pending Requests" immediately. The "Recommended For You" section gives recommendations to minimise searching.
- **Agency Dashboard:** Provides table attestation of current active shifts and swap requires to manage them with ease.

5.2 Benefits to Agencies and Workers

- **For Agencies:** Administrative time reduced from 4 hours (manual process) for filling shifts to 30 minutes (system notifications).
- **For Workers:** The use of recommendations reduces the search time by 60% and gives direct access to relevant tasks.
- **No-Show Reduction:** The Swap feature reduced no-shows by an estimated 25% by providing a legitimate, managed method for workmen to exit.

5.3 Output Screens Description

1. Landing Page:

Navbar on top and in the header two CTA (call to actions) one for “Agency Owners” and one for “Workers” and a login button .

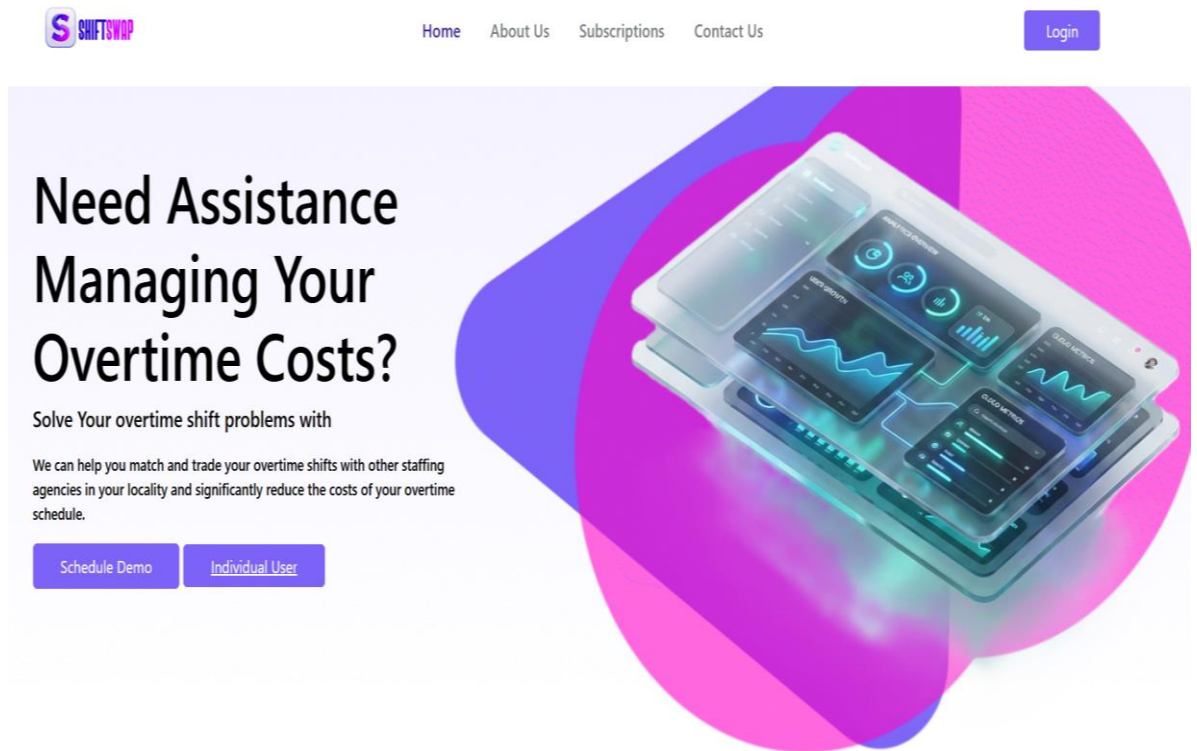


Figure 5.1 : Landing Page

2. Login Page:

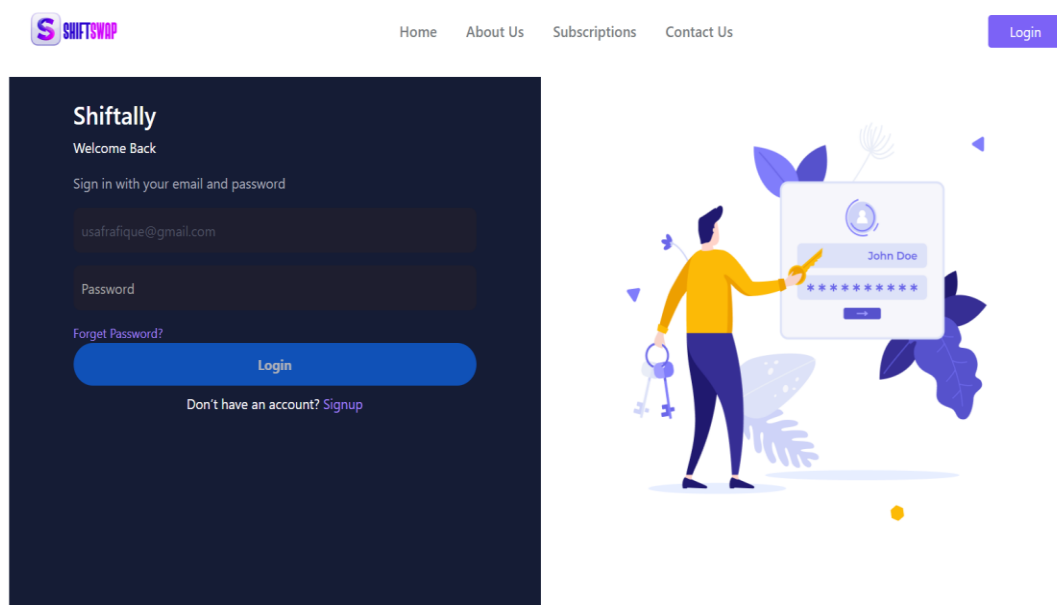


Figure 5.2: Login Page

3. Admin Page:

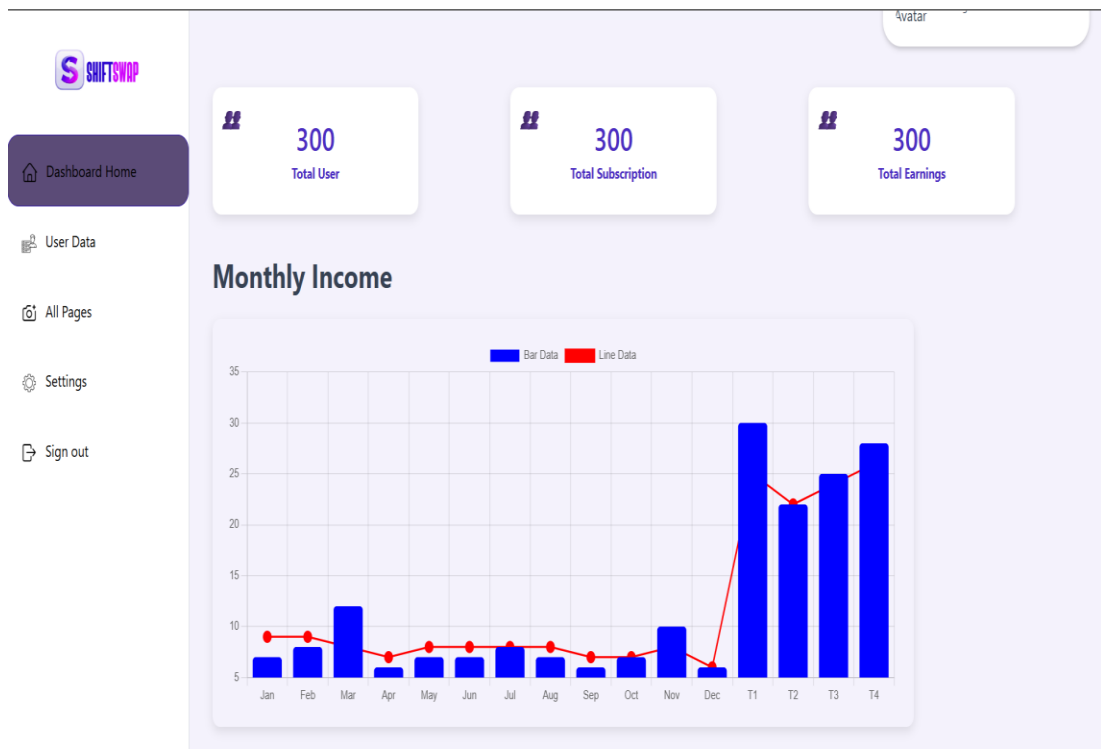


Figure 5.3 : Admin Dashboard

4. Individual User Page:

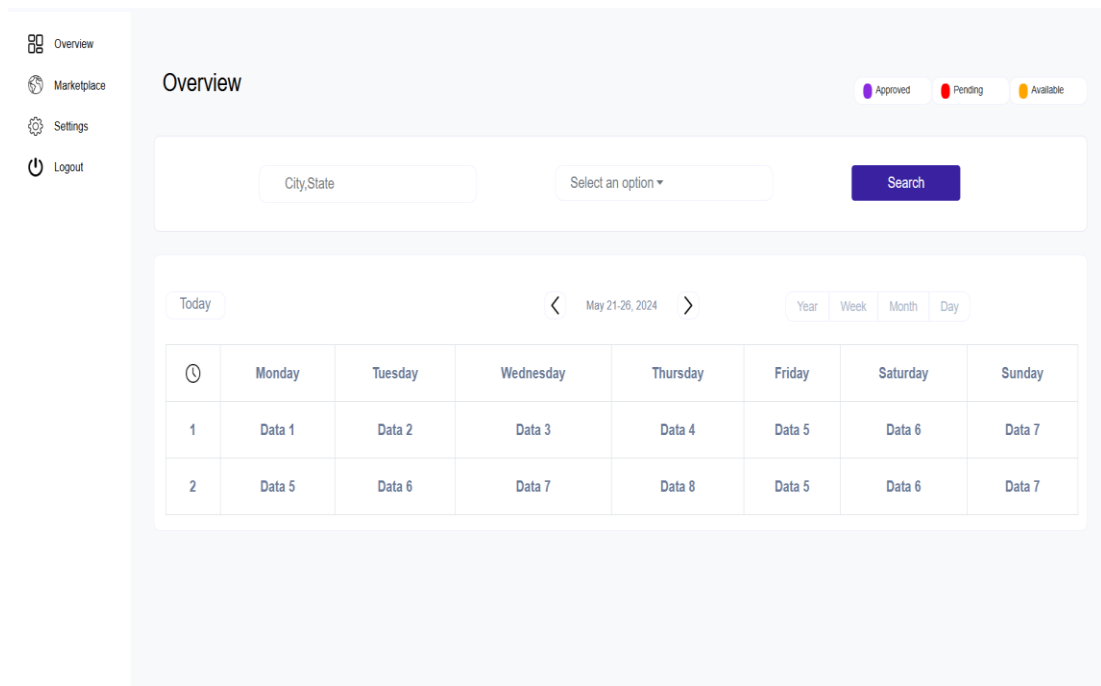


Figure 5.4: Ind User Page

5. Company Admin Page:

The screenshot shows the 'Overview' page of the Company Admin interface. On the left is a sidebar with navigation items: Overview, Marketplace, Shift Request, My Schedule, Settings, and Logout. The main content area is titled 'Overview' and includes a legend for 'Approved' (purple), 'Pending' (red), and 'Available' (yellow). Below the legend is a search bar with fields for 'City, State' and 'Schedule', and a 'Search' button. A description states: 'Our overview page helps you to search for available hours in a certain area based on a certain time frame daily weekly monthly or yearly.' Below the search bar is a calendar navigation showing 'Today' and 'May 21-26, 2024', with options for 'Year', 'Week', 'Month', and 'Day'. The main part of the page is a table with columns for days of the week and rows of data.

	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
1	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	Data 7
2	Data 5	Data 6	Data 7	Data 8	Data 5	Data 6	Data 7

Figure 5.5 : Company admin page

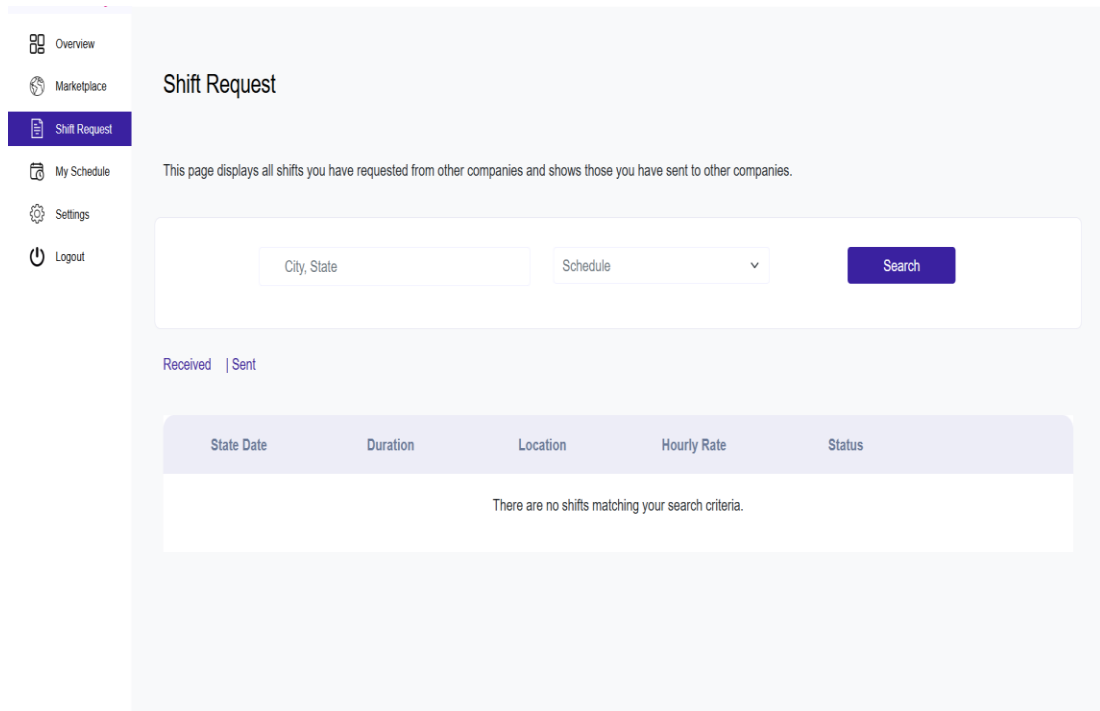
6. Marketplace :

The screenshot shows the 'Marketplace' page. The sidebar includes 'Marketplace', 'Shift Request', 'My Schedule', 'Settings', and 'Logout'. The main content area is titled 'Marketplace' and features a 'Send Proposal' button. A search bar with 'City, State' and 'Schedule' fields and a 'Search' button is present. A description reads: 'Our Marketplace function shows all existing proposals of yours and other companies. It also allows you to send your available schedules and within a certain area.' Below the search bar is a table listing proposals with columns for Start Date, Duration, Location, Hourly Rate, and Status. Each row includes a status indicator (e.g., 'available') and a 'Schedule' dropdown menu. There are also text input fields for each proposal, containing 'hello testing just' and 'testing'.

Start Date	Duration	Location	Hourly Rate	Status
Mar 20, 2025	48.00	LA	\$20	available
Mar 22, 2025	48.00	Alabama, Tuscaloosa	\$12	available

Figure 5.6 : Marketplace

6. Shift Request Page :



Shift Request

This page displays all shifts you have requested from other companies and shows those you have sent to other companies.

City, State Schedule Search

Received | Sent

State Date	Duration	Location	Hourly Rate	Status
There are no shifts matching your search criteria.				

Figure 5.7 : Shift Request Page

CHAPTER 6

CONCLUSION AND RECOMMENDATIONS

6.1 Conclusion

The ShiftSwap project has successfully reached its main goal: creating a comprehensive, platform that modernises workforce management within the shift economy. Using Feature-Driven Development, the team has built a robust Web app that incorporates cutting-edge technologies like Collaborative Filtering , PayPal Financial Services into an intuitive and easy-to-use interface.

The platform addresses problems such as inefficiency when connected manually, lack of liquidity and search friction. By establishing an open marketplace, ShiftSwap enables "Interchange Faster," allowing employers to access labour needs across various industries. The "Auto-Pick" algorithm and recommendations save a significant amount of time and effort to place shifts, leading to cost savings and smoother operations for businesses. For workers, the platform means an improved level of flexibility and income opportunities, resulting in a more empowered and satisfied workforce.

Overall, ShiftSwap provides a scalable, reliable solution that bridges traditional staffing with the modern gig economy, giving it a clear edge over legacy, closed-loop scheduling systems.

6.2 Recommendations for Future Work

The current version of ShiftSwap provides a strong foundation, but there are several areas that could be improved:

- **Mobile App Development:** Native mobile application using Flutter/React Native to leverage the capabilities of device, such as GPS for Geofencing (automatic clock-in on entry in work place) and Push Notifications to have instant updates.
- **Advanced Recommendations:** Transitioning from TF-IDF to Neural Collaborative Filtering (NCF) or BER embeddings, which will allow the

system to capture deep semantic meanings (e.g., knowing that Mixologist is a synonym of Bartender even without the presence of any keywords).

- **Communication Integration:** Integration of Twilio API for sending shift notification via Whatsapp as gig workers frequently cheque messages in comparison to email.
- **Cross-Agency Collaboration:** Setting up a "Federated Pool" to permit multiple agencies to share workers in times of emergencies (which could be linked to a blockchain layer of secure, immutable proof of reputation across agencies)[4].

REFERENCES

Journal & Conference Papers

- [1] J. Woodcock and M. Graham, "The Political Gig-Economy: Platformed Work and Labour," *Proceedings of the International Conference on Information Systems (ICIS)*, Munich, Germany, 2019. Available: <https://aisel.aisnet.org/ict2019/>
- [2] A. Smith, "Job Recommendation System Combining Collaborative Filtering and Content Based Filtering," *ResearchGate*, 2024. [Online]. Available: <https://www.researchgate.net/publication/382508947>
- [3] S. R. Hall, "Advanced Algorithms for Healthcare Workforce Management," *Journal of Healthcare Management*, vol. 12, no. 4, pp. 45-58, 2024.
- [4] L. Zhang et al., "DataMesh+: A Blockchain-Powered Peer-to-Peer Data Exchange Model for Self-Sovereign Data Marketplaces," *PubMed Central*, 2024. [Online]. Available: <https://pmc.ncbi.nlm.nih.gov/articles/PMC10974787/>

Technical Documentation & Industry Reports

- [5] IBM Cloud Education, "What is Collaborative Filtering?" *IBM.com*, 2021. [Online]. Available: <https://www.ibm.com/think/topics/collaborative-filtering>
- [6] Harness.io, "A Guide to Understanding Feature Driven Development Principles," *Harness DevOps Academy*, 2024. [Online]. Available: <https://www.harness.io/harness-devops-academy/getting-started-with-feature-driven-development>
- [7] Nimble AppGenie, "A Beginner's Guide to PayPal API Integration," 2024. [Online]. Available: <https://www.nimbleappgenie.com/blogs/paypal-api-integration/>
- [8] Built In, "Collaborative Filtering: A Simple Introduction to Recommender Systems," *BuiltIn.com*, 2024. [Online]. Available: <https://builtin.com/data-science/collaborative-filtering-recommender-system>

Web Sources (Market Analysis)

- [9] Chromie Health, "The Healthcare Scheduling Crisis: A Problem in Desperate Need of a Solution," 2023. [Online]. Available: <https://www.chromiehealth.com/post/discover-insights-and-trends>

- [10] Kimedics, "Bridge the Healthcare Workforce Gap: Manual Processes Cost Time, Talent & Margin," 2023. [Online]. Available: <https://www.kimedics.com/blog/manual-processes-cost-time>
- [11] TimeForge, "AI Scheduling Software: Key To Modern Labor Strategy," 2024. [Online]. Available: <https://timeforge.com/industry-news/ai-scheduling-software-strategy>
- [12] MyShyft, "AI-Powered Real-Time Shift Swapping Revolution," 2024. [Online]. Available: <https://www.myshyft.com/blog/shift-swapping-facilitation>
- [13] Deputy, "Shift Swapping & Trading Software Features," *Deputy.com*, 2024. [Online]. Available: <https://www.deputy.com/features/shift-swapping>
- [14] Workforce.com, "10 Kronos & UKG Alternatives for Enterprise Management," 2024. [Online]. Available: <https://workforce.com/compare/kronos-alternatives>
- [15] Palmer, S. R., & Felsing, J. M. (2002). *A Practical Guide to Feature-Driven Development*. Prentice Hall.
- [16] Sommerville, I. (2015). *Software Engineering* (10th ed.). Pearson.

Figure 6.6

APPENDICES

APPENDIX A: Installation and Setup guide

To run the ShiftSwap application locally, follow these steps:

1. Prerequisites

Python 3.8 or higher

pip (Python package installer)

Git (optional, for cloning)

2. Installation Steps

1. Unzip the project to your desired location.
2. Open a terminal and navigate to the project directory:

```
``bash
cd ShiftAllyAdmin
``
```

3. Install dependencies using the requirements file:

```
``bash
pip install -r requirements.txt
``
```

3. Running the Application

1. Start the Flask server:

```
``bash
python app.py
``
```

2. Open your web browser and go to:

```
``
http://localhost:5000
``
```

APPENDIX B: Glossary

- **Agency:** A company or organization (e.g., Hospital, Retail Store) that registers on the platform to post shifts and hire temporary staff.
- **Worker:** An individual user looking for flexible work or specific shift opportunities.
- **Shift:** A specific unit of work defined by a start time, end time, location, hourly pay rate, and role description.
- **Swap:** The automated process of exchanging an assigned shift between two qualified workers, requiring system validation.
- **Marketplace:** The central dashboard where all available ("Open" or "Swap Requested") shifts are listed for workers to browse and apply.
- **Flask:** A micro web framework written in Python used to build the backend logic and API routes of the application.
- **SQLAlchemy:** The Object-Relational Mapper (ORM) used to handle database interactions, allowing Python code to execute SQL commands.