SWIFTFIX: HOME SERVICES WEB APP

# PROJECT REPORT

Submitted in partial fulfillment of the requirements of the degree

### BACHELOR OF ENGINEERING IN INFORMATION TECHNOLOGY

By

SANSKRUTI ADAP - 01

#### SOHAM BABSHETYE - 06

PARTH DHANORKAR - 24

AMIT KORADE - 53

Guide:

## Prof. Reshma Malik



Department of Information Technology

**Thadomal Shahani Engineering College**

Adv. Nari Gursahani Marg, TPS III, Off Linking Rd, Bandra West, Mumbai, Maharashtra 400050

**University of Mumbai**

(Academic Year 2023-24)

# CERTIFICATE

This is to certify that the Mini Project entitled **“SWIFTFIX”** is a bonafide work of **Amit Korade (53), Parth Dhanorkar (24), Soham Babshetye (04), Sanskruti Adap (01)** submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of **“Bachelor of Engineering”** in

**“Information Technology Engineering”.**

|  |  |  |
| --- | --- | --- |
| **Prof. Reshma Malik** | **Dr. Mukesh Israni** | **Prof. G.T. Thampi** |
| Project Guide | Head of Department | Principal |

# Mini Project Approval

This Mini Project entitled **“SWIFTFIX”** by **AMIT KORADE (17), PARTH DHANORKAR (30), Sanskruti Adap (01), SOHAM BABSHETYE (29)** is approved for the degree of **Bachelor of Engineering** in **Information Technology Engineering.**

### Examiners

**1………………………………………**

#### (Internal Examiner name & Sign)

**2…………………………………………**

#### (External Examiner name & Sign)

Date:

Place: Mumbai

## ACKNOWLEDGEMENT

This project has consumed a significant amount of time and resources, and we would like to thank everyone who has contributed to its success. We would like to express our deepest gratitude to the project head, **Prof. Reshma Malik** for her time, patience and direction, as well as for inspiring us throughout. Without their guidance and support, this project would not have been possible. Their expertise, insights, and encouragement were invaluable in helping us navigate the various challenges and complexities of this project.

We would also like to extend our sincere thanks to the many research papers that we consulted throughout the course of this project. The insights and ideas presented in these papers were instrumental in helping us develop a deeper understanding of the subject matter and in shaping the direction of our research.

We are also grateful to our colleagues and friends who provided us with valuable feedback and support throughout the project. Their encouragement and insights were invaluable in helping us refine our ideas and in pushing us to achieve our best work.

## ABSTRACT

SwiftFix, a home services platform, aims to address challenges homeowners face in finding reliable and reasonably priced services such as plumbing, cleaning, electrical repairs, and maintenance. SwiftFix offers a user-friendly application that aggregates service providers, enabling quick booking of trusted professionals with transparent pricing. Our project leverages technology to streamline the search for home services, providing a one-stop solution for homeowners in need of trusted, accessible, and fairly priced professionals. Key features include background-checked service providers, transparent pricing, flexible subscription options, and easy booking.

|  |  |
| --- | --- |
| **CONTENTS** | **Page No.** |
| **Chapter 1: Introduction** | **1** |
| 1.1 Introduction | 1 |
| 1.2 Motivation | 1 |
| 1.3 Problem Statement & Objectives | 1 |
| 1.4 Organization of The Report | 2 |
| **Chapter 2: Literature & Survey** | **3** |
| 2.1 Survey of The Existing System | 3 |
| 2.2 Limitation of Existing System or Research Gap | 3 |
| 2.3 Mini Project Contribution | 4 |
| **Chapter 3: Proposed System** | **5** |
| 3.1 Proposed System | 5 |
| 3.2 Architecture/Framework | 5 |
| 3.3 User Flow | 6 |
| 3.4 Database Structure | 8 |
| **Chapter 4: Conclusions and Future Work** | **9** |
| **References** | 10 |

## CHAPTER 1: INTRODUCTION

### Introduction

SwiftFix is a home service aggregation platform developed to facilitate access to trustworthy and affordable professionals for essential home services. It simplifies service selection and booking, ensuring that homeowners can easily connect with qualified professionals without the hassle of comparing multiple providers.

### Motivation

The motivation for SwiftFix stems from the growing need for a reliable, user-centric platform that brings transparency and efficiency to the home service industry, addressing the common pain points of unclear pricing, inconsistent service quality, and limited availability.

### Problem statement and objectives

Finding reliable home services at fair prices can be challenging. Homeowners frequently struggle to locate trustworthy service providers, resulting in wasted time and delayed maintenance.

Our primary objectives in developing this application are:

1. **User-Friendly Interface:** A seamless interface for booking services.
2. **Transparent Pricing:** Consistent and fair pricing models.
3. **Trusted Professionals:** Ensure safety and reliability through background checks.
4. **Convenience & Flexibility:** Variety in services and subscription plans.

### Organization of the report

This report is divided into four chapters. The first chapter covers the project's introduction, problem description, motivation for the topic, and objectives. The second chapter is a review of the literature. It contains all of the research effort done on this topic. This chapter covers all you need to know about studying current systems and learning new technologies. The third chapter describes the suggested system that will be employed in this project. This chapter contains screenshots of the project's block diagram, methodologies, hardware, and software. All materials referred and utilized in the development of this project are included in the references section.

## CHAPTER 2: LITERATURE SURVEY

### Survey of existing system

Several platforms have emerged to simplify the process of finding reliable home services, with notable examples including *Urban Company* and *SweepSouth*. These platforms allow users to book various home services, from cleaning and repairs to beauty and wellness treatments. They provide access to a network of professionals who are vetted for quality, allowing users to browse and book appointments directly through their mobile apps or websites.

* **Urban Company**: Urban Company offers a wide range of home services, such as plumbing, electrical repairs, and home cleaning. Its strengths lie in its large service provider network and strong brand presence in regions like India, Singapore, and Australia. Urban Company also emphasizes safety by conducting background checks on service professionals and setting standard prices, which helps build trust with users.
* **SweepSouth**: This South African-based service platform focuses on household cleaning services and aims to connect users with cleaners who can be scheduled on demand. SweepSouth is popular for its flexible scheduling and convenient booking interface. It operates primarily in South Africa but is expanding into other African markets. It also focuses on the needs of working professionals, making home cleaning services affordable and easily accessible.

### Limitation of existing system or research gap

While Urban Company and SweepSouth provide valuable services, several limitations leave gaps for improvement. These limitations highlight the need for platforms that are more customizable, transparent, and versatile for users with varied needs.

1. **Limited Personalization**: Both platforms provide generalized services without offering personalized options based on specific user requirements. For instance, users might need tailored maintenance plans or customized service bundles, which are not consistently available on these platforms. This limitation can leave users feeling that services are one-size-fits-all, lacking customization to meet their unique needs.
2. **Pricing Transparency and Flexibility**: While Urban Company offers standardized pricing, it can still lack flexibility in terms of discounts or package deals. Additionally, the lack of dynamic pricing in response to demand or user subscription models can make regular maintenance services costly. SweepSouth, on the other hand, focuses mainly on cleaning services, and its limited service scope does not fully meet the diverse needs of homeowners seeking comprehensive home services.
3. **Niche Service Limitations**: Both Urban Company and SweepSouth do not cater extensively to specialized services, such as elder care, pet care, or personal chef services, which are becoming increasingly important for urban households. Expanding into these niche areas can better address the evolving lifestyle needs of modern users.
4. **Limited User Engagement and Support Features**: Existing systems offer basic customer support, which often involves a chatbot or contact center. However, there is room for improvement in terms of providing 24/7 support, multilingual assistance, and real-time updates on service providers’ availability. Many users would also benefit from a system that offers after-service feedback options to continuously enhance service quality.

### Mini project contribution

The SwiftFix platform seeks to address the gaps identified in existing systems by providing a more user-centric, flexible, and comprehensive home service solution. SwiftFix’s unique contributions can be summarized as follows:

1. **Personalized Services**: SwiftFix offers customized service packages and subscription plans that cater to specific needs, such as regular home maintenance, specialized elder care, and seasonal requirements like festive cleaning. This personalized approach ensures that users can tailor services according to their unique household needs, helping them avoid paying for services they do not require.
2. **Subscription-Based Plans**: SwiftFix provides three subscription models (Silver, Gold, and Platinum), each offering a different set of services and discounts. Users can select a plan based on their service frequency and needs, creating a cost-effective solution for those who require ongoing assistance, as opposed to a standard pricing model for every service booked.
3. **Enhanced Service Variety and Niche Offerings**: Unlike other platforms, SwiftFix includes a broad range of services that go beyond typical home maintenance, such as pet care, babysitting, and personal chef services. This service diversity allows users to address multiple needs within one platform, making it a true one-stop solution.
4. **Increased Transparency and Support**: With 24/7 chatbot support, SwiftFix ensures that users receive assistance any time they need it. Additionally, background checks and transparent pricing are essential to building user trust, while the multilingual feature (e.g., English and Hindi support) makes the platform accessible to a broader demographic.
5. **User-Centric Enhancements**: SwiftFix provides features like dark-light mode and periodic festive offers, making it easy and enjoyable for users to navigate and utilize the platform. These user-friendly enhancements foster engagement, helping users feel comfortable and valued on the platform.

## CHAPTER 3: PROPOSED SYSTEM

### Proposed system

SwiftFix is designed as a full-stack web application built using the MERN stack: **MongoDB**, **ExpressJS**, **ReactJS**, and **NodeJS**. This modern stack provides flexibility, scalability, and efficiency, making it ideal for an application focused on connecting homeowners with reliable service providers in real-time. By leveraging this stack, SwiftFix can offer a seamless user experience, ensuring smooth navigation, efficient data management, and rapid service booking and tracking.

The proposed system comprises three main user roles:

1. **Homeowners** seeking services.
2. **Service Providers** (SwiftStars) looking to connect with customers.
3. **Admin Users** for managing and monitoring the platform, ensuring quality control and compliance.

Key features of the proposed system include:

* **User Authentication and Role Management**: A secure login and registration system that differentiates between homeowners, service providers, and admins. User roles determine the level of access and functionality within the platform.
* **Real-Time Booking and Availability**: Users can view and book services in real time. Service providers update their availability, and homeowners receive live feedback on appointment status, promoting convenience and time management.
* **Pricing Transparency and Invoicing**: Transparent pricing based on service type and location, with a commission-based model that supports fair revenue distribution. After services are rendered, an automated invoicing system manages payment processes.
* **Reviews, Ratings, and Feedback Mechanism**: SwiftFix incorporates a feedback loop to maintain high standards of service by allowing users to review and rate providers, thus fostering trust within the community.
* **Multi-Language Support and 24/7 Assistance**: Multilingual functionality (initially in English and Hindi) and chatbot support provide an inclusive and accessible platform for diverse users.

SwiftFix is structured to adapt to user needs over time, with flexible subscription plans and seasonal promotions. This modular design allows for the easy addition of new features and services as market demands evolve.

### Architecture / Framework

##### Python Programming Language

The architecture of SwiftFix follows a modular, component-based approach using the MERN stack, which promotes efficient data handling, scalability, and secure, real-time data flows between users and the backend.

1. **Frontend**:

**ReactJS** powers the SwiftFix frontend, creating a responsive, user-friendly interface that offers seamless navigation and interaction. React’s component-based structure enables reusable UI elements, allowing for a consistent design and faster development.

**Redux** is utilized for state management, particularly to handle user sessions, booking status, and real-time updates, ensuring a fluid user experience even during high traffic.

* + **Responsive Design**: The frontend is built to adapt across various devices, ensuring accessibility on both desktops and mobile devices.

1. **Backend**:

**NodeJS and ExpressJS** form the core of the backend, acting as the server-side framework that processes user requests, manages data transactions, and handles the business logic. ExpressJS offers a lightweight and flexible solution for building the API endpoints that power SwiftFix’s features, such as user registration, booking, payments, and feedback.

* + **RESTful API Design**: The backend exposes a RESTful API that supports interactions between the frontend and the database. These APIs handle tasks like booking requests, managing provider profiles, retrieving services, and calculating fees or discounts in real-time.
  + **Authentication & Security**: Using JWT (JSON Web Tokens) for secure authentication, the backend ensures that only verified users can access sensitive information. Additionally, input validation and other security best practices are implemented to safeguard user data.

1. **Database**:

**MongoDB** serves as the primary database, chosen for its NoSQL, document-based model, which provides flexibility in storing and retrieving data, ideal for handling complex and evolving data structures such as user preferences, service provider profiles, and booking histories.

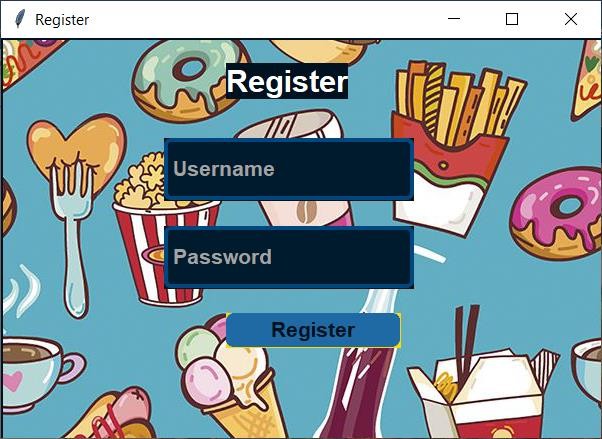
**Data Structure**: MongoDB’s collections and documents allow SwiftFix to store data in a format that is easily scalable and quick to query. Key collections include:

* + - **Users**: Stores details on homeowners, service providers, and admins, as well as roles, preferences, and subscription data.
    - **Services**: Contains information on the different services offered, along with pricing models, discounts, and seasonal promotions.
    - **Bookings**: Manages real-time bookings and statuses, allowing providers to update availability and users to track their service history.
    - **Reviews and Ratings**: Enables feedback tracking to maintain service quality and inform new users about reliable service providers.

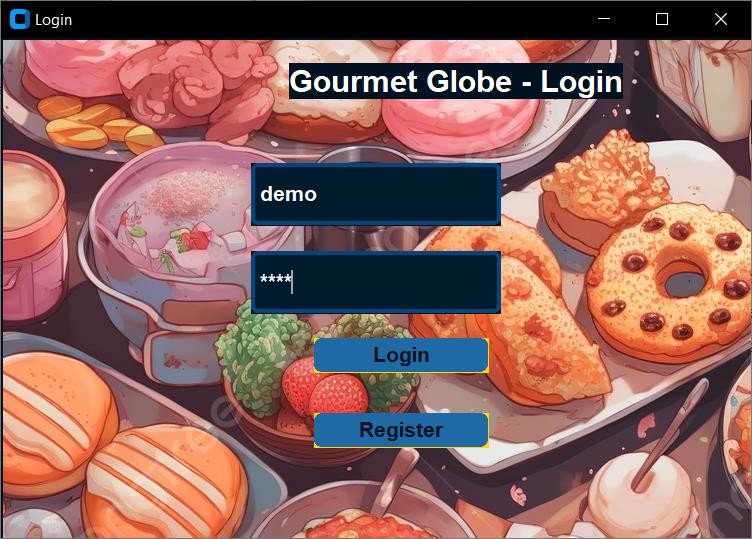
### User Flow

##### Login and Register

The user has to log in with a username and password, as shown in Figure 1. If the user doesn't have an account the user will create a new account by filling up some credentials like name, surname, username and then setting a password for the account.



**Fig 1: Login Page**

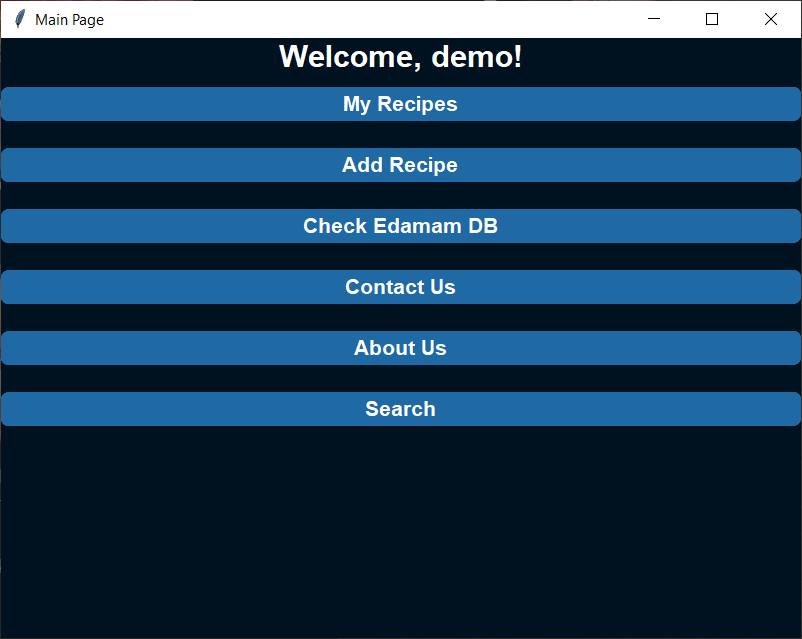


**Fig no. 2: Registration Page**

As shown in Figure 2, This is the registration page that the user is directed to if they do not have an account already made.

##### Home Page

Next, the user chooses between the various options and features available as shown in Figure 3.



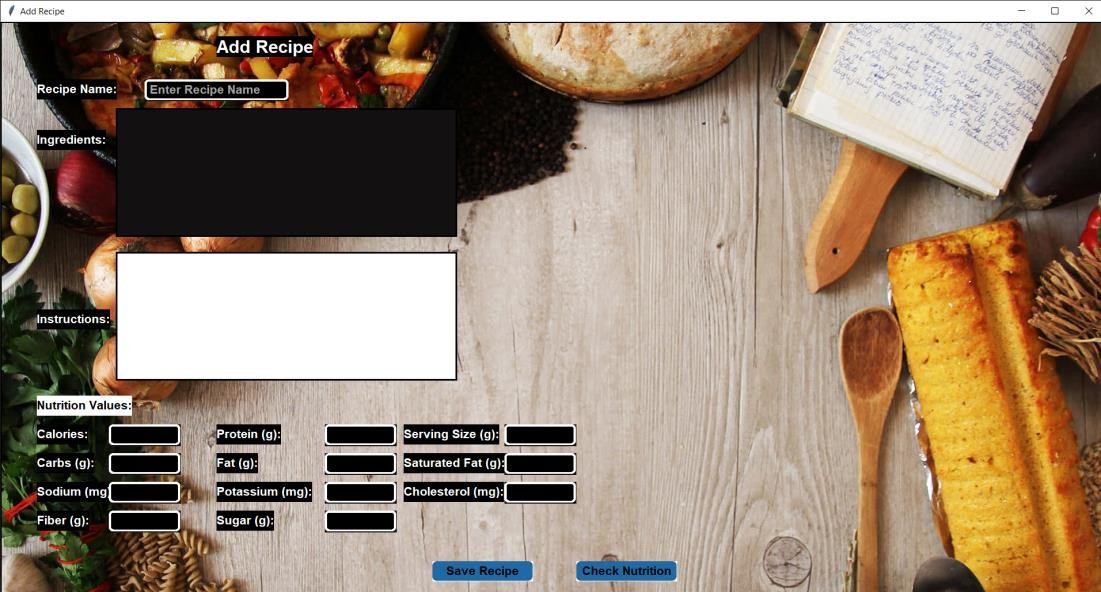
**Fig 3: Home Page**

##### Add Recipe Page

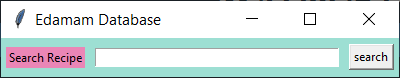
The add recipe page is for the user to add their own specialized recipes to the app. All these recipes would automatically get saved in the database and can be retrieved whenever the commands are passed

##### Edamam Database

The Edamam Database is an online database which is connected to the app using web scrapping in python. This feature helps the user to get any recipe available on the internet just by typing the name and then searching it.



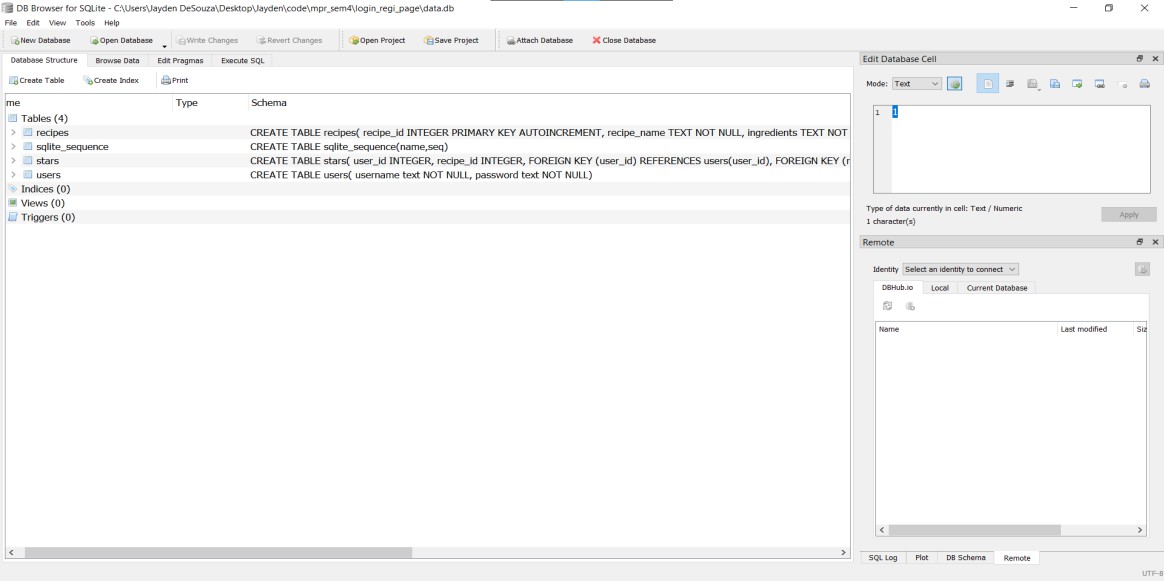
**Fig 4: Add Recipe**



**Fig 5: Web search recipes**

### DATABASE STRUCTURE

Shown in Figure 6, is a picture of our Database Structure that holds all the necessary information about the users and the recipes stored in the app, constantly being updated as new information is added.



**Fig 6: Database structure.**

**CHAPTER 4: CONCLUSION AND FUTURE WORK**

In conclusion, the Recipe App project has successfully achieved its objectives of creating a user-friendly application for accessing and managing recipes. By leveraging Python programming and database management techniques, we have developed a functional and intuitive software solution. The application allows users to add, delete, and search for recipes effortlessly, enhancing their cooking experience and organization of culinary resources.

Throughout the project, we have gained valuable experience in designing graphical user interfaces using tkinter, implementing database functionalities with SQLite, and structuring Python applications for practical use cases. The project serves as a testament to the versatility and power of Python in software development, particularly in the realm of creating desktop applications.

We could’ve added more features but that would require other languages and approaches.

Additionally, there are several avenues for future enhancement and expansion:

* User Authentication: Implement a user authentication system to allow multiple users to have personalized recipe collections and preferences.
* Recipe Ratings and Reviews: Introduce features for users to rate and review recipes, providing valuable feedback to the community and helping others discover popular recipes.
* Advanced Search Options: Enhance the search functionality with advanced filtering options, such as filtering by cooking time, dietary restrictions, or cuisine types.
* Ingredient Management: Incorporate functionality for managing ingredient inventory, including shopping lists, pantry tracking, and ingredient substitution suggestions.
* Better GUI: Even after using tkinter, other programming languages can provide a faster and detailed GUI than python programming language.

**REFERENCES**

* 1. This reference is used in the creation and connection of the database- [https://youtu.be/bmt1x\_Mjg-s?si=1R1p6D07i6dUleR](https://youtu.be/bmt1x_Mjg-s?si=1R1p6D07i6dUleRm)m

#### One of the food API’s that is used to collect and show users

multiple recipes- <https://www.edamam.com/>

* 1. Another food API used in the project- [https://spoonacular.com/food-ap](https://spoonacular.com/food-api)i