FIX FLEX

HELWAN UNIVERSITY

Faculty of Computers and Artificial Intelligence

Information System Department

FIX FLEX

A graduation project dissertation by:

- Ahmed Mostafa Tawfik 202000086
- Mostafa Yasser Faroq 202000920
- Menna AlaaEldein AbdElmoen 202000947
- Aisha Ahmed Fathy 202000486
- Basmala Yasser Mohamed 202000200
- Esraa Raafat Hekal 202000113

Submitted in partial fulfilment of the requirements for the degree of Bachelor of Science in Computers & Artificial Intelligence at the Information System Department, the Faculty of Computers & Artificial Intelligence, Helwan University

Supervised by:

Helal Ahmed Soliman

Table of Content

Chapter 1

Introduction

1.1 Overview

What is FIX FLEX?

FIX FLEX is a platform that connects people who need tasks done with individuals who are willing to complete those tasks for pay.

It operates like traditional freelancing platforms such as Upwork, Fiverr, and Freelancer, where users can post tasks or jobs and taskers make offers on these tasks.

The user then selects the most suitable tasker for the job. However,

the main difference between fixing and traditional freelancing platforms lies in their primary objectives.

While platforms like Upwork primarily focus on digital services such as programming, writing, and design,

FIX FLEX core mission is to facilitate the completion of a wide range of everyday tasks and services.

FIX FLEX specializes in connecting people who need tasks like (plumbing, electricity, painting, cleaning, moving, and more)

with handymen who are willing to perform these tasks for pay.

For post a task, creating a job listing is straightforward. Upload a detailed description of your task, specify the category it falls under, set your budget, and add any relevant images or documents. With a wide array of categories including home improvement, you're sure to find the right professional for your needs.

The Tasker can set offers that users can choose from.

For freelancers and businesses, FIX FLEX offers a robust platform to showcase your skills. Create a compelling profile highlighting your expertise, experience, and previous work. Browse through a variety of tasks, submit quotes, and communicate directly with potential clients.

The platform emphasizes transparency and trust. All users can view and compare ratings and reviews, ensuring a reliable and quality experience. Payments are secure and straightforward, handled efficiently within the app.

FIX FLEX offers mobile apps for both iOS and Android devices, allowing users to access the platform conveniently

from their smartphones or tablets. This enhances the accessibility of the platform, enabling users to post tasks, make offers, and communicate with each other on the go

Whether you're looking to get a task done or to offer your skills, FIX FLEX is your go-to platform for seamless, efficient, and trustworthy service exchanges.

1.2 Objectives

- **Accessibility for All**: FIX FLEX primary goal is to be an inclusive platform, welcoming users from various walks of Egypt. Whether you're a professional seeking more work, or an individual in need of assistance with a task, FIX FLEX is designed to cater to your needs. We aim to bridge the gap between demand and supply in the service sector, ensuring easy access for everyone.

- **Support for Varied Needs**: Understanding that needs can range from simple household tasks to complex professional services, FIX FLEX is committed to providing a comprehensive range of categories. This ensures that no matter what your requirements are, whether it is repair or refurbishment, you will find a skilled professional to help you.

- **Upholding Privacy and Confidentiality**: We place a high value on the privacy and confidentiality of our users. Personal

information and transaction details are securely handled and protected, ensuring a safe and trustworthy environment.

- **Ensuring Quality and Reliability**: To maintain the highest standards, FIX FLEX implements a robust system of ratings and reviews. This not only helps in building trust but also ensures that only qualified and reliable professionals are connected with those needing services.

- **Fostering a Community of Support:** By providing a platform where individuals can seek help and professionals can offer their services, FIX FLEX aims to foster a sense of community and mutual support. We believe in creating a network where users can share their experiences, help each other grow, and contribute to a more connected and supportive soc

1.3 Purpose

Diversity in services At FIX FLEX, it provides many services that the customer needs, with high quality, so that the customer can find what he needs in one place. **the speed** In FIX FLEX, it works to quickly respond to the request when it occurs due to the speed of its repair and ensures its presence in all governorates in Egypt and its rapid arrival.

1-4 Scope

- **Broad Geographic Reach**: FIX FLEX aims to start by assisting people across various locations and regions. The project's primary focus is on inclusivity, reaching out to individuals in need across different areas and governorates.
- **Verification and Transparency**: ensure the authenticity of the cases and build trust with donors, the platform will support campaigns with photos. This approach is intended to provide donors with confidence and assurance that their contributions are reaching the intended recipients.
- **Open Platform:** FIX FLEX will be an open platform where connecting people who need tasks like (plumbing, electricity, painting, cleaning, moving, and more)

1-5 General Constraint

Understanding the limitations and constraints of a project is crucial for its successful execution and management. Here are the general constraints for the FIX FLEX project:

- **Time Constraint**: The project is subject to a specific timeline, which includes deadlines for each phase and the final rollout date. Adhering to this schedule is vital to ensure timely completion and launch. The time constraint encompasses not only the duration of tasks but also the allocation of resources and coordination among different project components.

- **Scope Constraint**: This defines the boundaries of the project, including its goals, deliverables, features, and functions. The scope outlines the tasks required to complete the project and sets the parameters within which the project must be developed. It is essential to maintain a clear and focused scope to avoid scope creep, which can lead to delays and additional costs.

- **Cost Constraint:** The budget of the project is a critical constraint, encompassing all financial resources required for timely and effective completion within the defined scope. However, the cost constraint is not limited to monetary expenditure alone; it also includes resource allocation, such as time, labor, and material resources. Effective cost management

is crucial to ensure that the project does not exceed its financial limitations.

- **Collecting Raw Data**: Gathering the necessary data for the project poses its own set of challenges. These include:

- Concept of Data Collection: Understanding what data is needed, why it is needed, and how it will be used is fundamental to the data collection process.

- Types of Data: Identifying the types of data and their sources is crucial for informed decision-making and project planning.

- Issues to be Considered for Data Collection: Addressing potential issues such as data privacy, accuracy, and relevance is essential for effective data collection.

- Methods of Primary Data Collection: Determining the most suitable methods for gathering primary data is vital for obtaining accurate and useful information.

By acknowledging and managing these constraints, the FIX FLEX project can be effectively guided towards successful completion, ensuring that it meets its intended goals within the set time frame, scope, and budget, and with the necessary data in hand.

Chapter 2
Planning and Analysis

2.1 Project planning

2.1.1 Feasibility Study

- **Financial Feasibility**

  The system will follow free software standards. Bug fixing and maintenance tasks will have a cost associated with it and in the initial stage the potential market space will be local and will be of free standards.

- **Technical Feasibility**

  Project FIX FLEX is a web application and mobile application. The main technologies and tools that are associated with FIX FLEX are:

  - **JavaScript**
  - **UI material**
  - **React**
  - **NodeJS**
  - **Flutter**

  Each of the technologies are freely available and the technical skills required are manageable.

2.2 Need for the new application

Like any program, it has a competitor, and there are many programs with the same ideas Our competitor is Task Rabbit , Handy, TAKL

TaskRabbit

Advantage:

• TaskRabbit conducts background checks on its taskers, ensuring users have access to trusted and reliable professionals.

Disadvantage:

• TaskRabbit charges a service fee on each transaction, which can make it more expensive compared to other platforms.

Handy

Advantage:

• Handy offers next-day availability for various home services, providing quick and efficient service for urgent tasks.

Disadvantage:

• Users have reported that the services offered can be less customizable compared to other platforms, which might not suit specific needs.

TAKL

Advantage:

• Takl provides transparent, pre-defined pricing for tasks, which helps users understand costs upfront without surprises.

Disadvantage:

• Takl's service availability can be limited in some regions, which might restrict access for some users.

Why FIX FLEX Is Better?

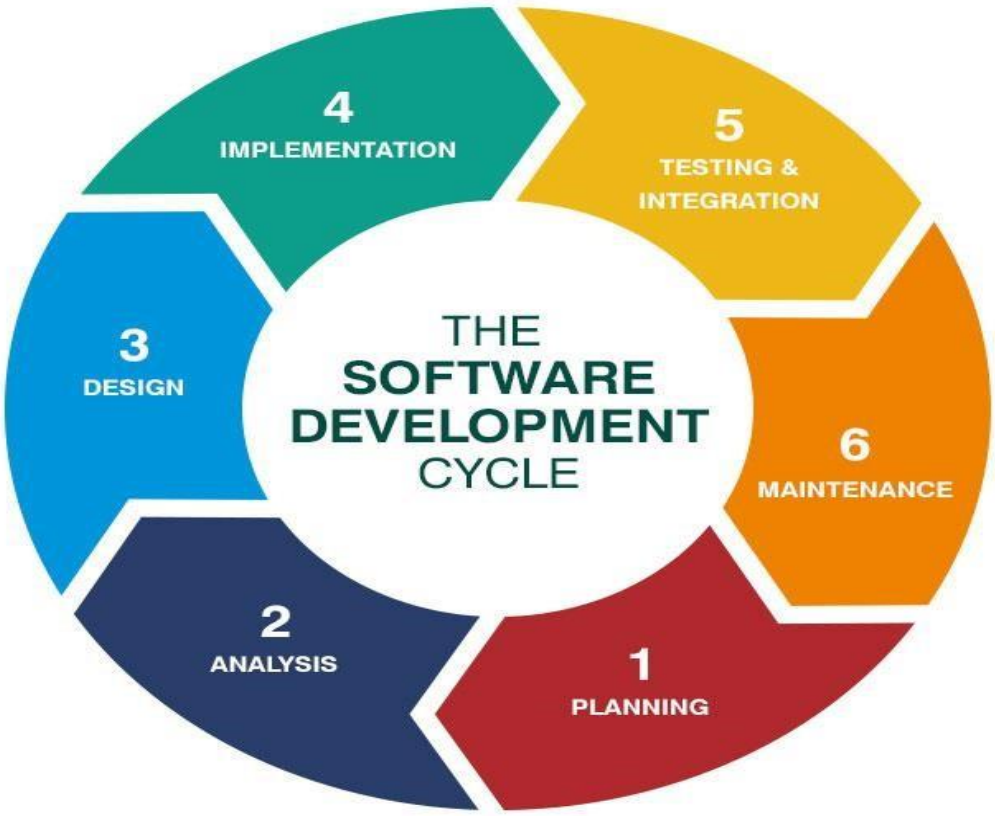
FIX FLEX does not charge any service fees, unlike TaskRabbit which has service fees for each transaction

Users can be both taskers and users with one account, offering more flexibility without managing multiple accounts

Our Al Assistant simplifies task posting by automatically filling in the details based on a brief description, saving time and effort

System Development Life Cycle

We followed system development life cycle approach in developing this project with its 7 phases, planning, analysis, design, development, testing, deployment and maintenance.



Methodology

We followed Agile analysis and design methodology in developing, this project and developed all related UML diagrams, represented in this document later.

2.3 Analysis of the system

2.3.1 User requirements

- This explains what the user will get through the site or application that you are using or the user what will it does inside the system?
- Often referred to as user needs, describe what the user does with the system, such as activities which users should be able to do.
- User requirements are documented in the User Requirements Document (URD) using narrative text.

Our FIX FLEX will consist of the following users:

1 - The User:

- He can create an account for the platform by registering his basic data (first name, last name, email, password, mobile number, address).
- He can then log in (email, password).
- upload his own profile image.
- update his own profile details.
- Users can post tasks they need help with, providing details (title, category, details, location, date, time, budget).
- The user receives offers from the Tasker and can accept or reject it.
- Users can track the status of their posted tasks and receive notifications on updates.
- Users and taskers can communicate in real-time through the built-in chat system.
- Users can rate taskers and leave reviews based on their experience.
- Users can payment cash or online payment.

1.1-Tasker:

It can do everything that the user does in addition to:

- Uploading the type of work.
- Can browse available tasks and choose the ones they want to complete.
- Make an offer to the user.

3-Admin:

- Delete task.
- Manage his profile.
- Update task.
- Add category.
- Delete category.
- Update category.
- See all users in the system.

2.3.2 System Requirements:

- what is the software that I use for build FIX FLIX is Visual Studio Code.
- the database used in system by MongoDB.
- programming languages are used in FIX FLEX is JavaScript, UI material, React, NodeJS, Flutter, Dart.

2.3.3 Functional Requirements

-Sign Up:

- The user logs in by (first name, last name, email, password, mobile number, address).
- B-The Data will be saved in the database.
- The user becomes registered in the system.

-Login User:

- A-The user tries to enter the site by Email and Password if the data is correct, if the data is incorrect there appears to him error message.
- B- Now the user can use the site.
- C-Every User can edit his profile.
- D-Upload profile image

-Login Admin:

- Log in to the system by email and password.
- B-Can Update his profile.
- C-Can delete any user or tasker.
- D- Can See all the details that are specific to a particular post.
- E- Can See all requests and can delete them.

-Post a Task:

- They log in or sign up if they haven't already.
- After successfully login or sign up, they click on "Post a Task".
- They select the category of the task (e.g., cleaning, gardening, IT support).
- They describe the task they need help with.
- They add additional task details such as location and due date.
- They set a budget for the task.
- They preview the task details to ensure accuracy.
- If the task details are correct, they post the task.

-Browse Task:

- They browse available tasks on the platform.
- If they find a task of interest, they view the task details.
- If they're interested in the task, they make an offer to the task poster.
- If the offer is accepted, they proceed to complete the task.
- After completing the task, they request payment from the task poster.
- If payment is received, the task is considered completed successfully.
- If payment is not received, they follow up for payment.

- Chatting:

- After the user and tasker logs in to the system.
- The user can communicate with the tasker in real-time through the built-in chat system.
- Who uses this:
    - Any user registered in the system wants to talk to Tasker

-Task Status Tracking:

- After the user logs in to the system and posted task.
- Users can track the status of their posted tasks and receive notifications on updates.
- Who uses this:

The user who posted the post

-Ratings and Review

- After the user logs in to the system and posted task and carry out his mission.
- Users can rate tasker and leave reviews based on their experience
- Who uses this:
    - Any user to whom Tasker provided a service

2.3.4 Non- Functional Requirements

- Performance Requirements:

-The system supports the use of multiple users at the same time.

- The database should be normalized to prevent redundant data and improve the performance.

- The database should be distributed to prevent outages.

- We will make database backup several times.

- Availability Requirements:

The database may crash at any certain time due to virus or operating system failure. Therefore, it is required to take the database backup.

- Security Requirements:

- The system should be completely Consistent and Secure.
- Keep specific history data sets.
- Communication needs to be restricted when the application is validating the user.

-Usability:

- user can achieve their goal easily.
- Speed of performing tasks for the user.
- Ease and simplicity of design

2.4 Advantages of the application

- First, FIX FLEX aims to bring together home repair services in one easy-to-access place
- The system will not focus on one service, but will focus on more than one service it provides
- Communication between user and tasker
- FIX FLEX provides service throughout Egypt

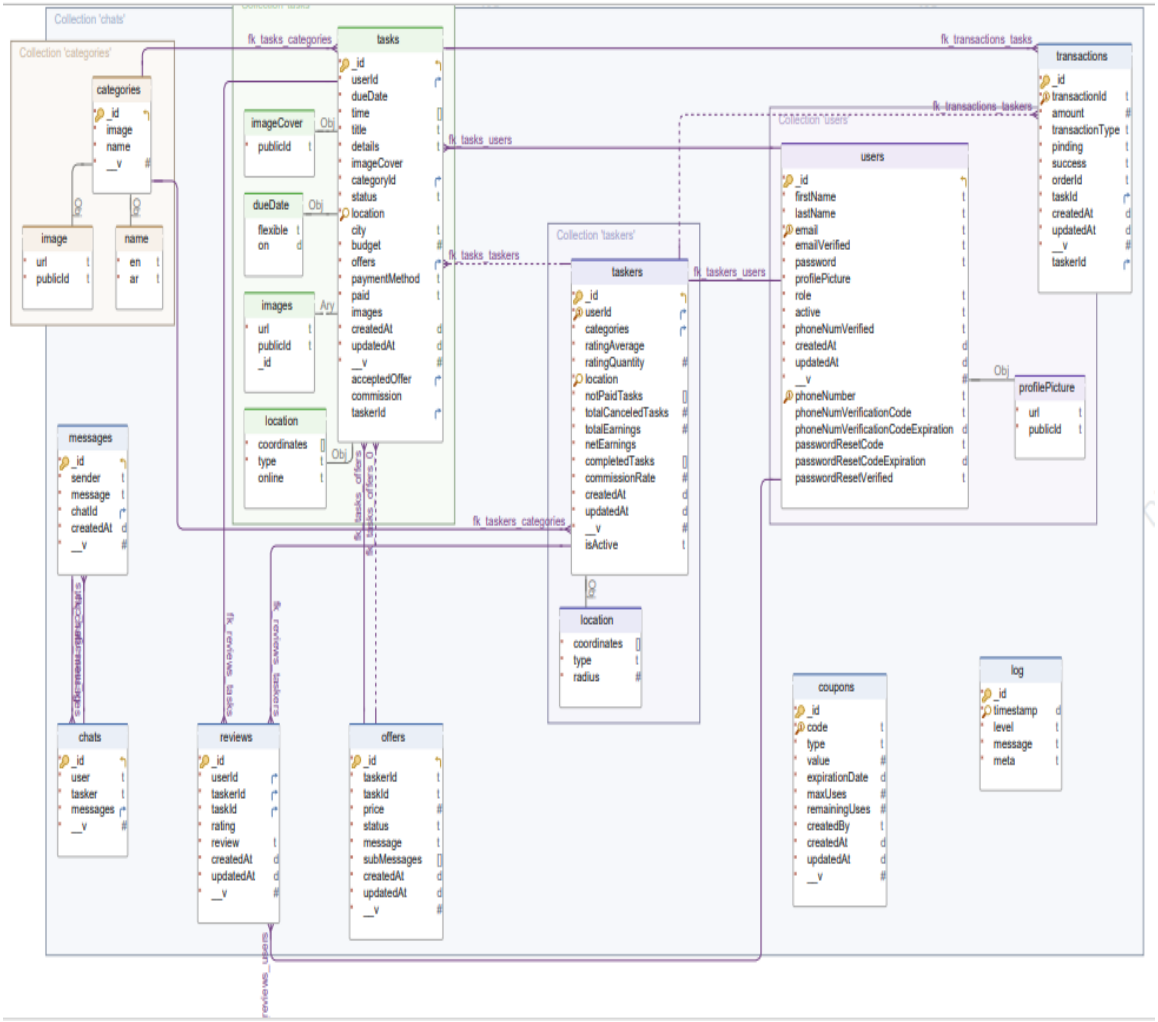
2.5 Risk and Risk Managements

Risks:

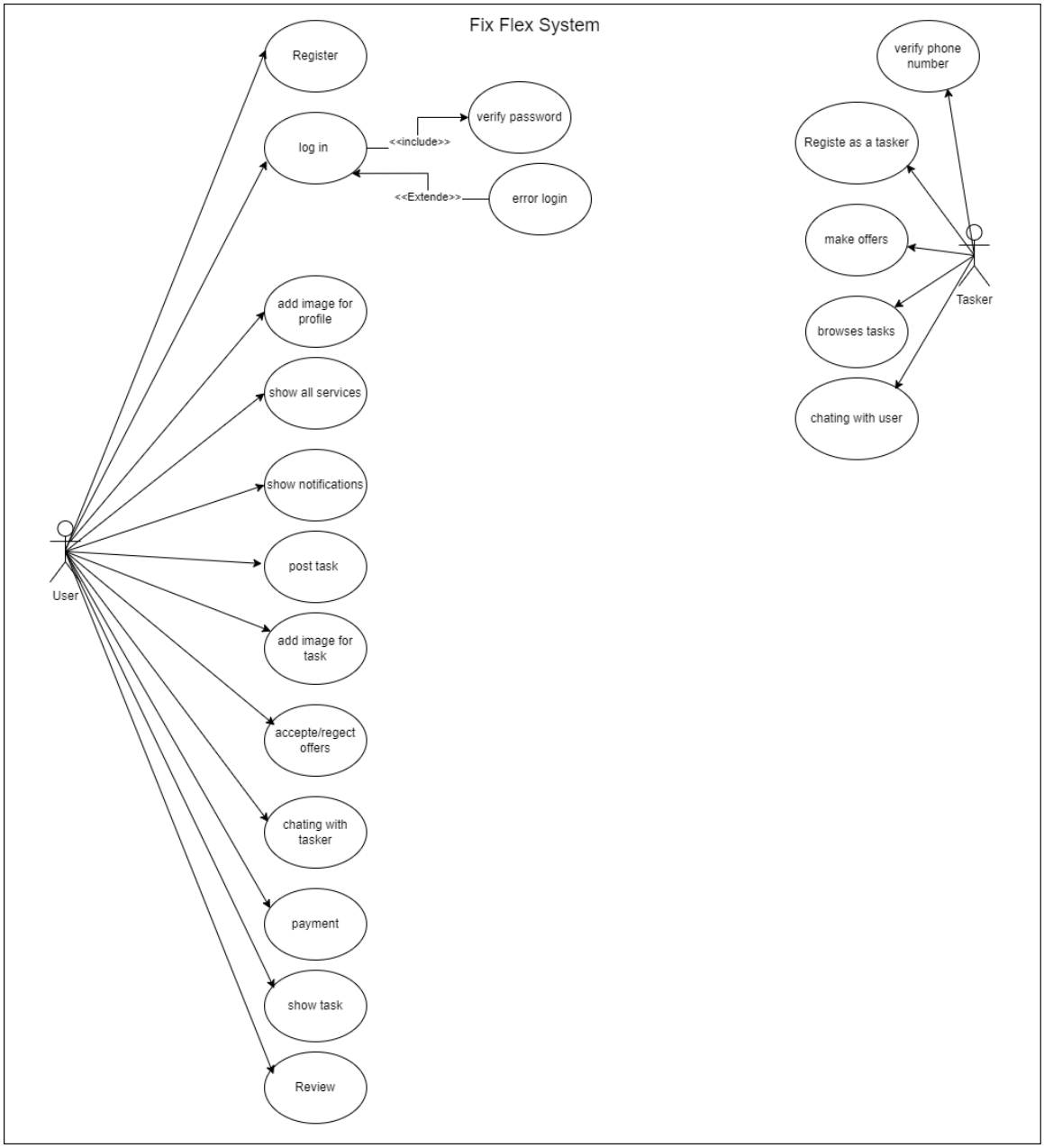
1. Not taking trust from customers
2. Great responsibility due to providing many services

Chapter 3
Software Design

3.1 Design of database

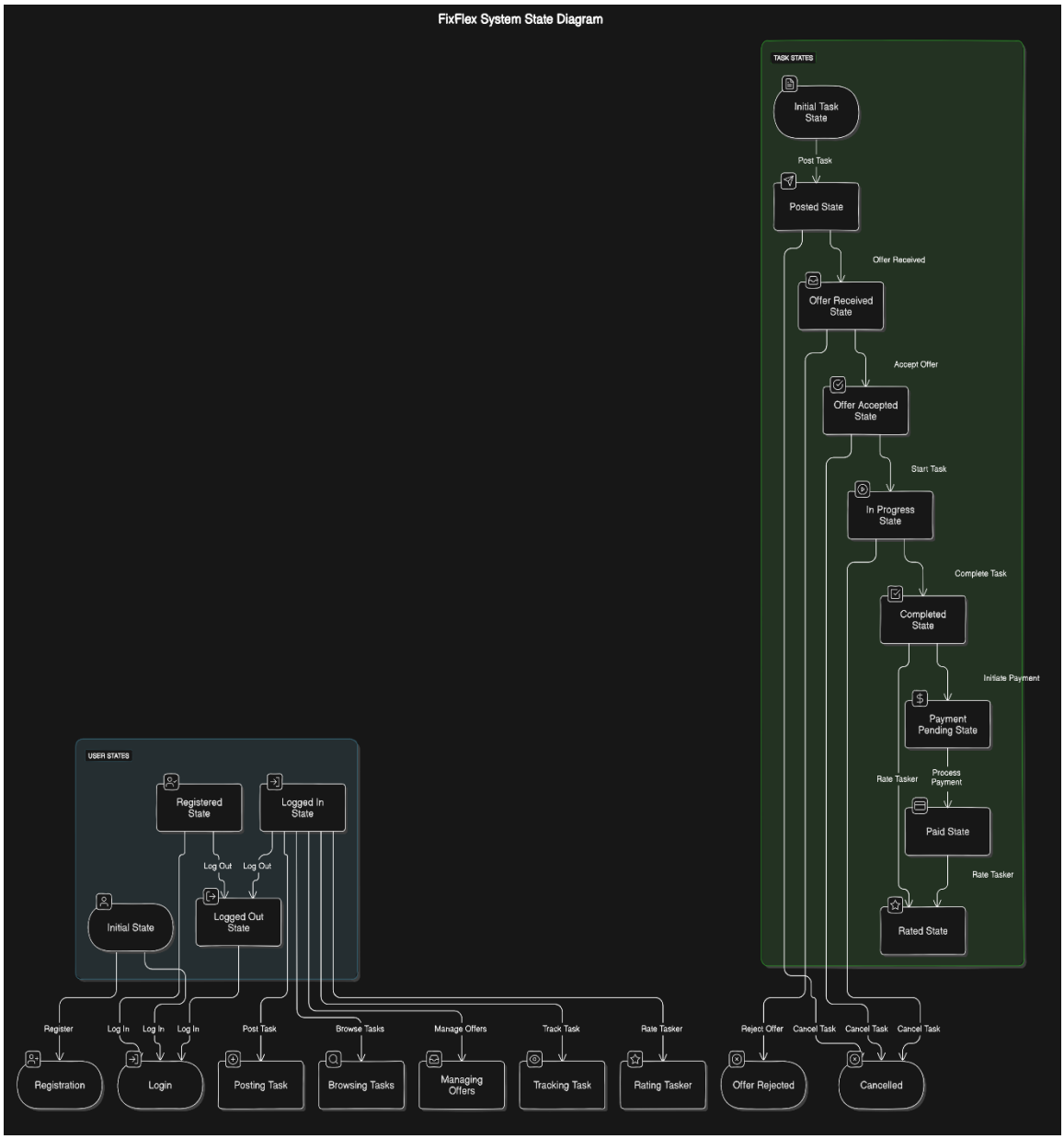


3.2 Use Case Diagram

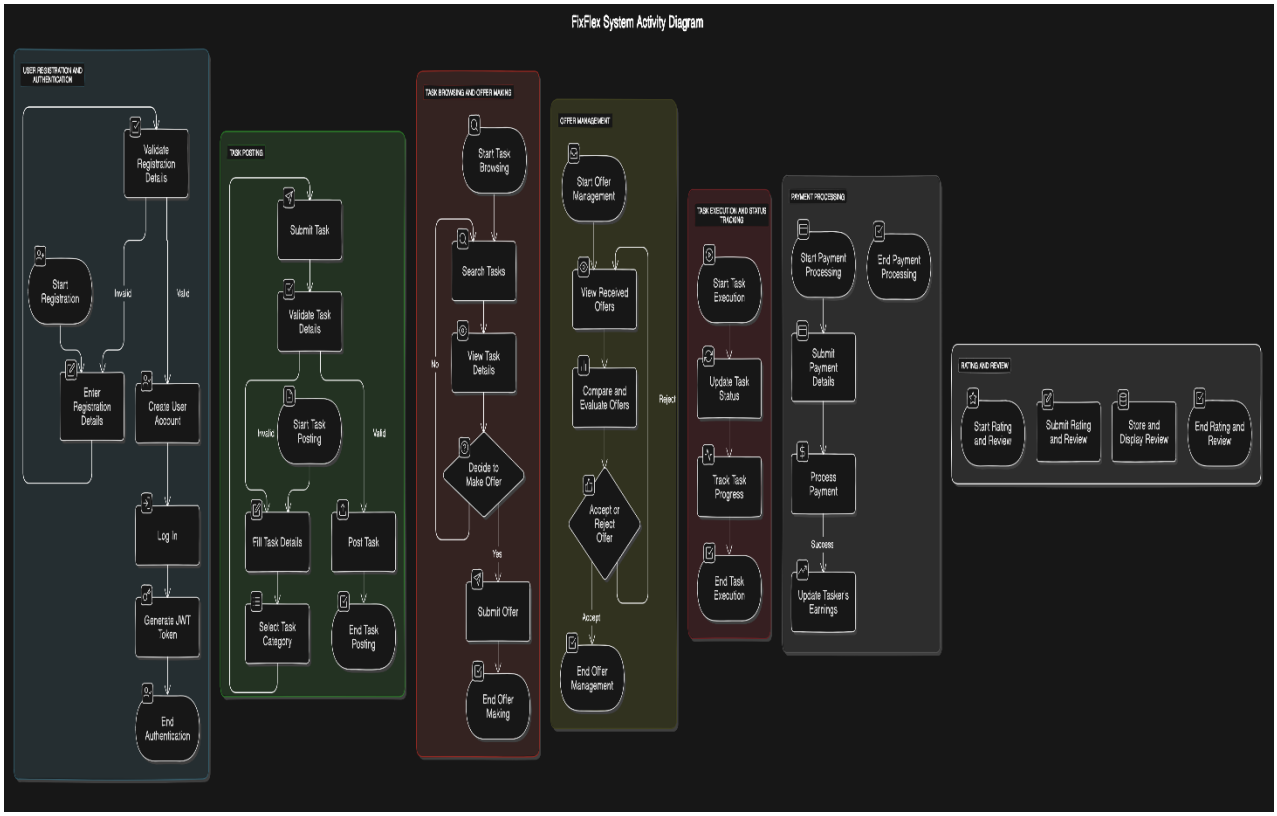



delete category
add category
get task by id
delete task by id
update task by id
see all users in system
manage profile
updete category
add image for category
admin

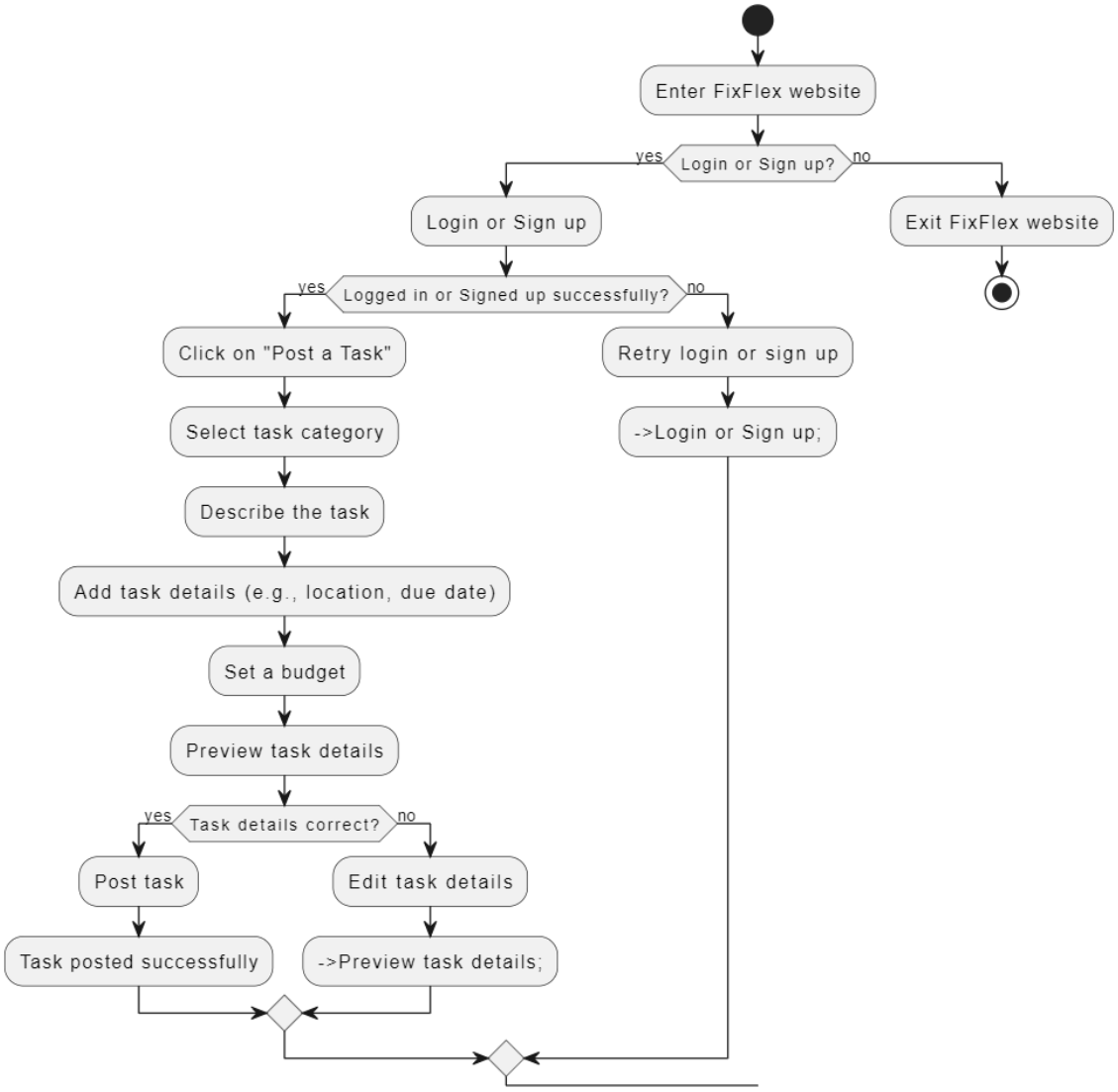
3.3 State Diagram



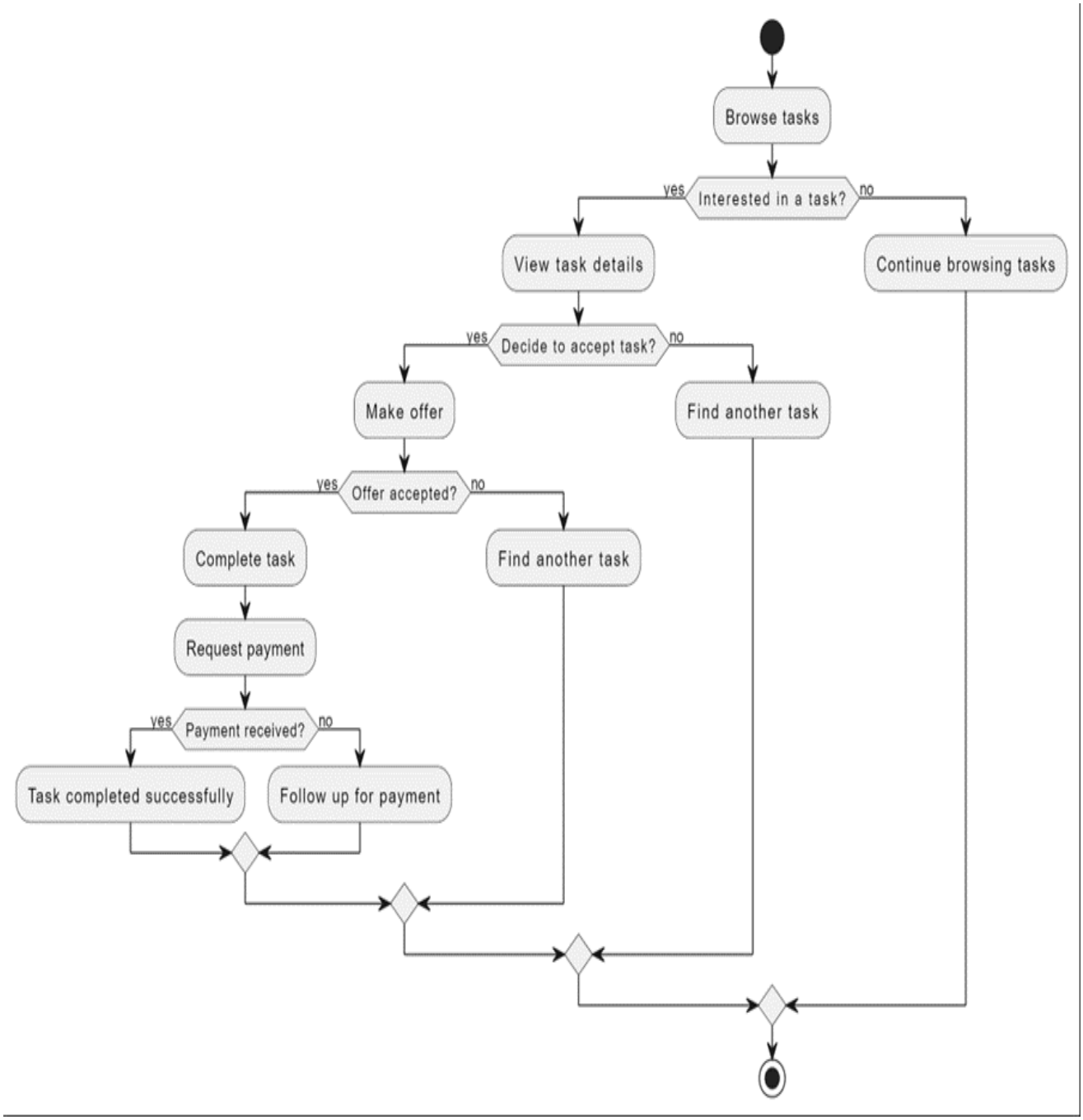
3.4 Activity Diagram



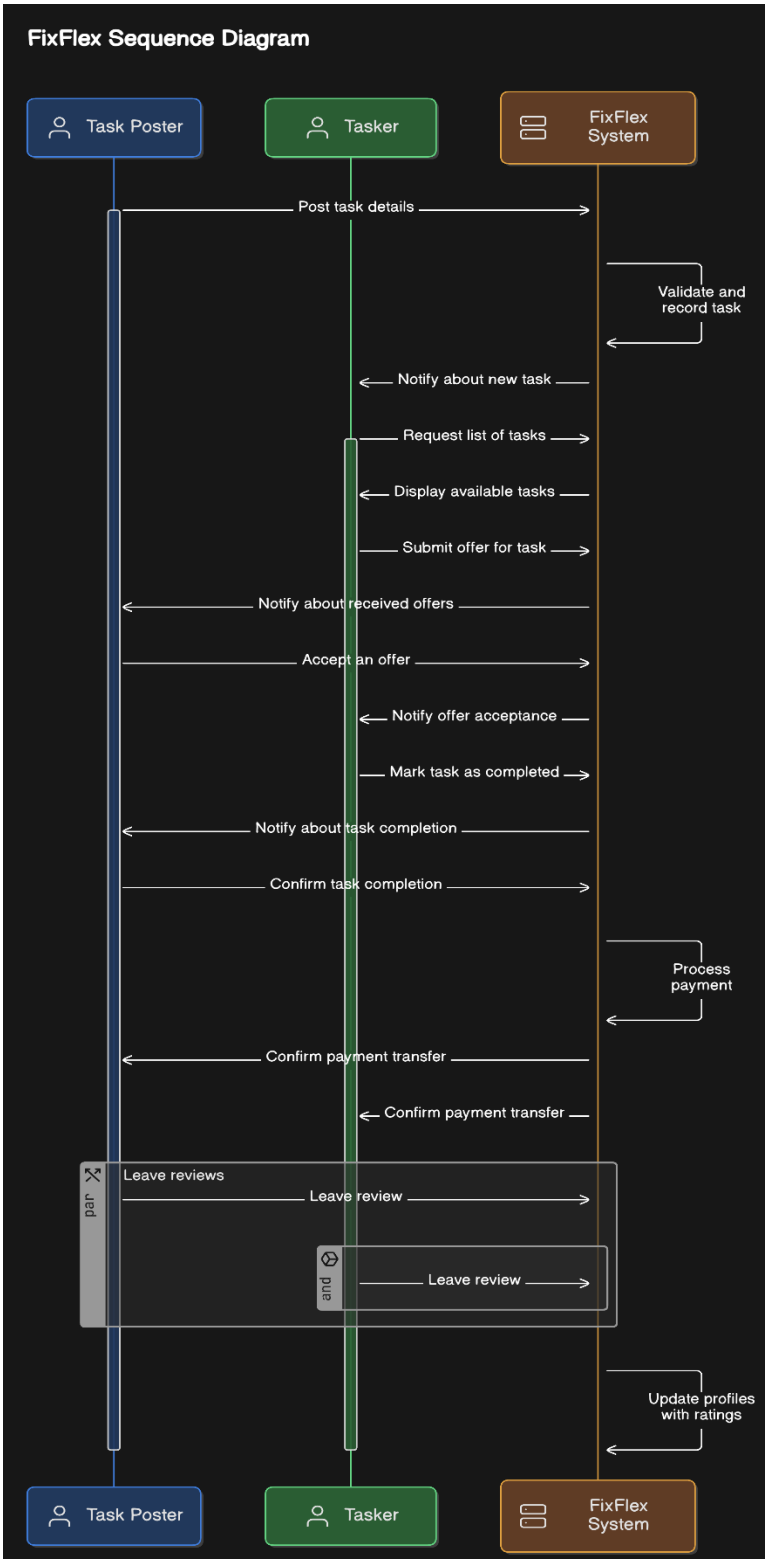
- Post Task

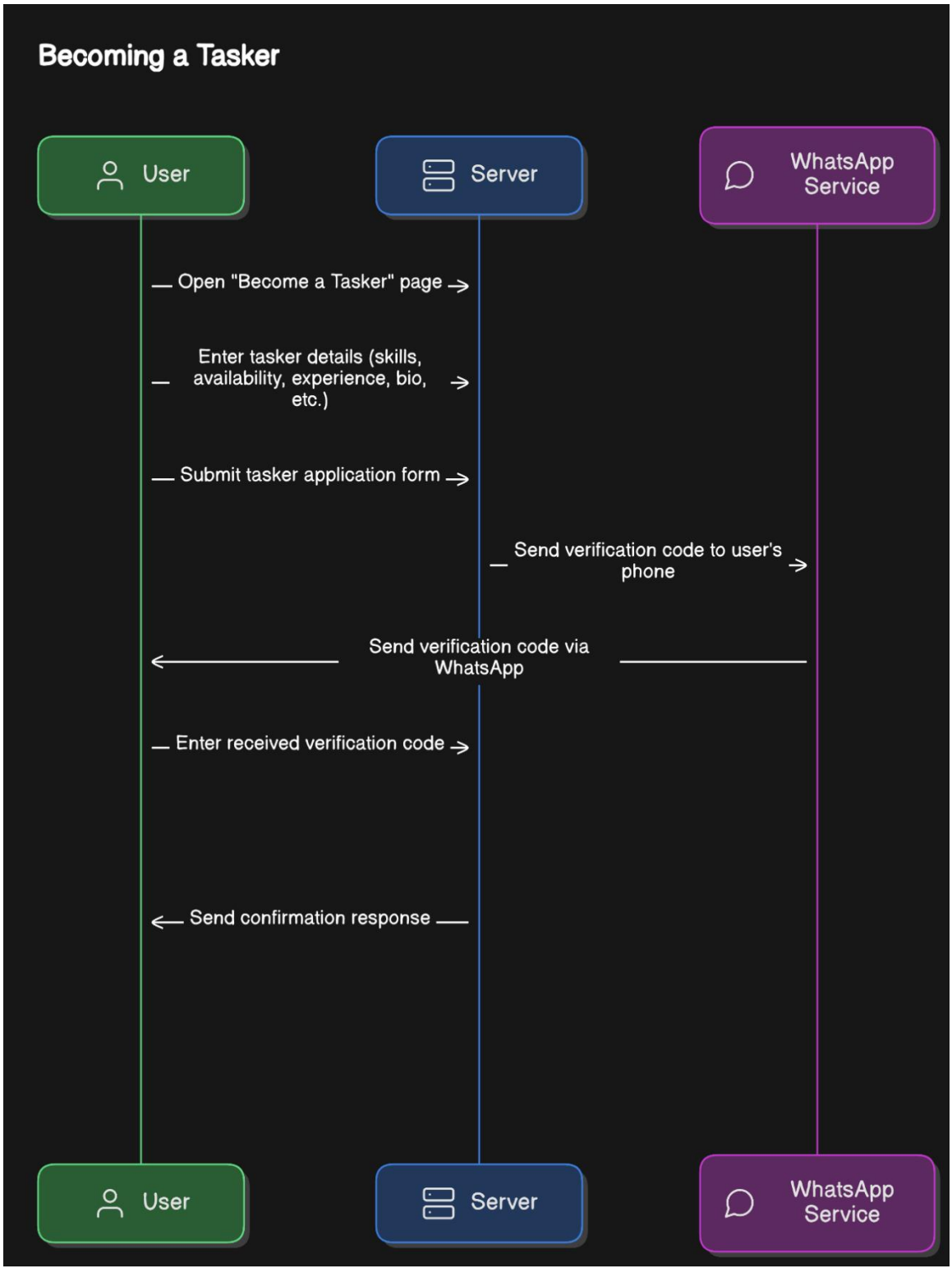

Enter FixFlex website
Login or Sign up?
yes
no
Login or Sign up
Exit FixFlex website
Logged in or Signed up successfully?
yes
no
Click on "Post a Task"
Retry login or sign up
Select task category
->Login or Sign up;
Describe the task
Add task details (e.g., location, due date)
Set a budget
Preview task details
Task details correct?
yes
no
Post task
Edit task details
Task posted successfully
->Preview task details;

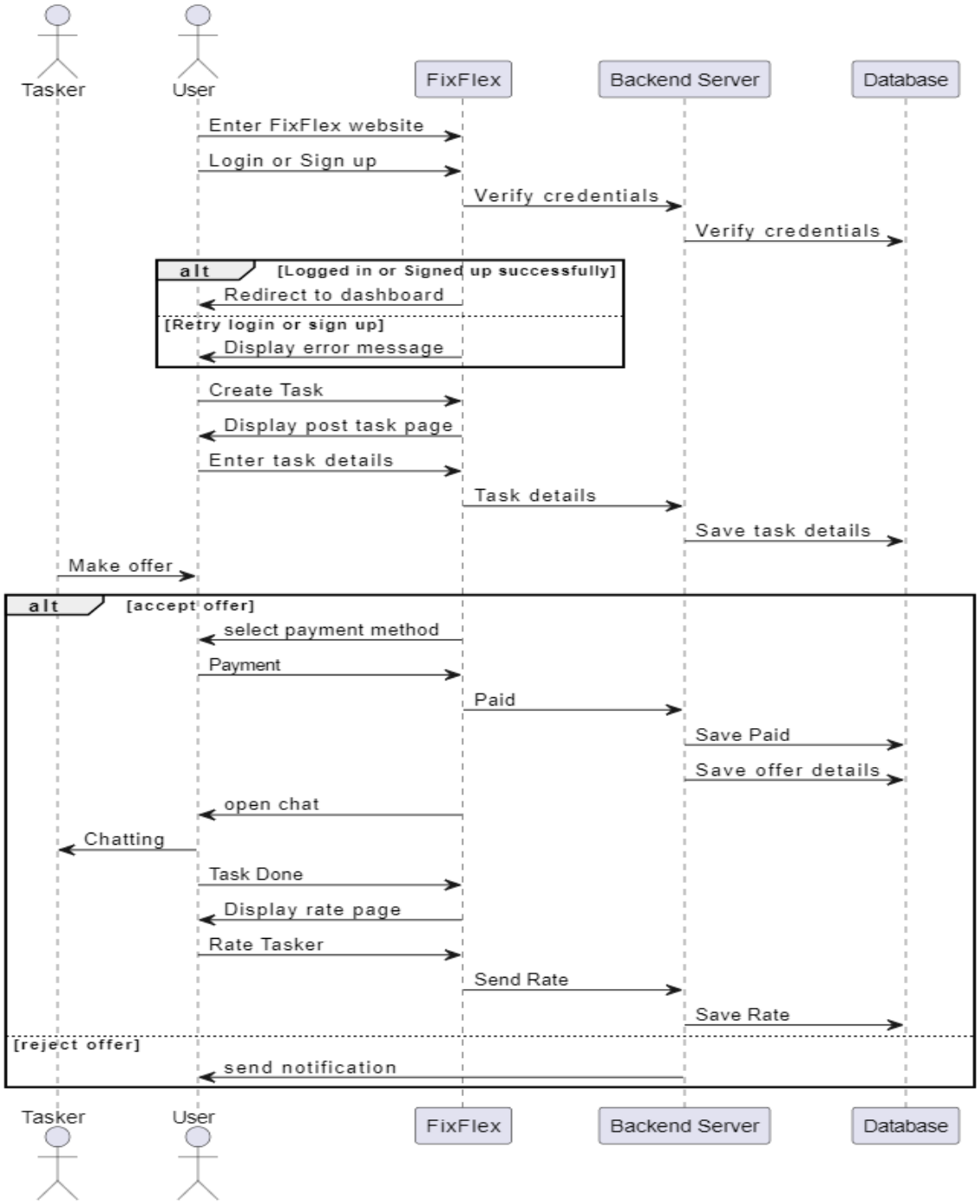
- **Make offer**


Browse tasks
Interested in a task?
yes
no
View task details
Continue browsing tasks
Decide to accept task?
yes
no
Make offer
Find another task
Offer accepted?
yes
no
Complete task
Find another task
Request payment
Payment received?
yes
no
Task completed successfully
Follow up for payment

3.5 Sequance Diagram



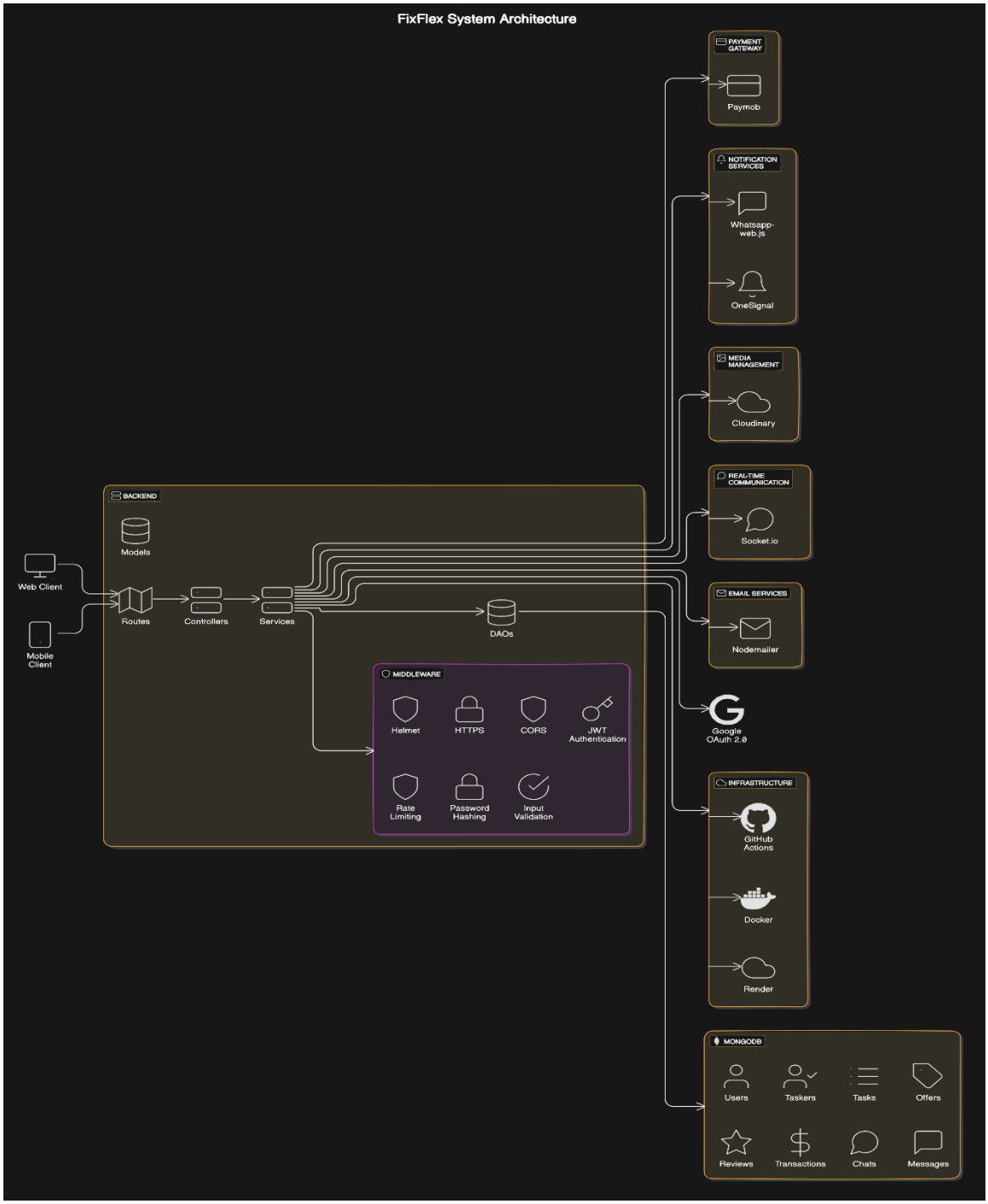

Becoming a Tasker
User
Server
WhatsApp Service
Open "Become a Tasker" page
Enter tasker details (skills, availability, experience, bio, etc.)
Submit tasker application form
Send verification code to user's phone
Send verification code via WhatsApp
Enter received verification code
Send confirmation response
User
Server
WhatsApp Service

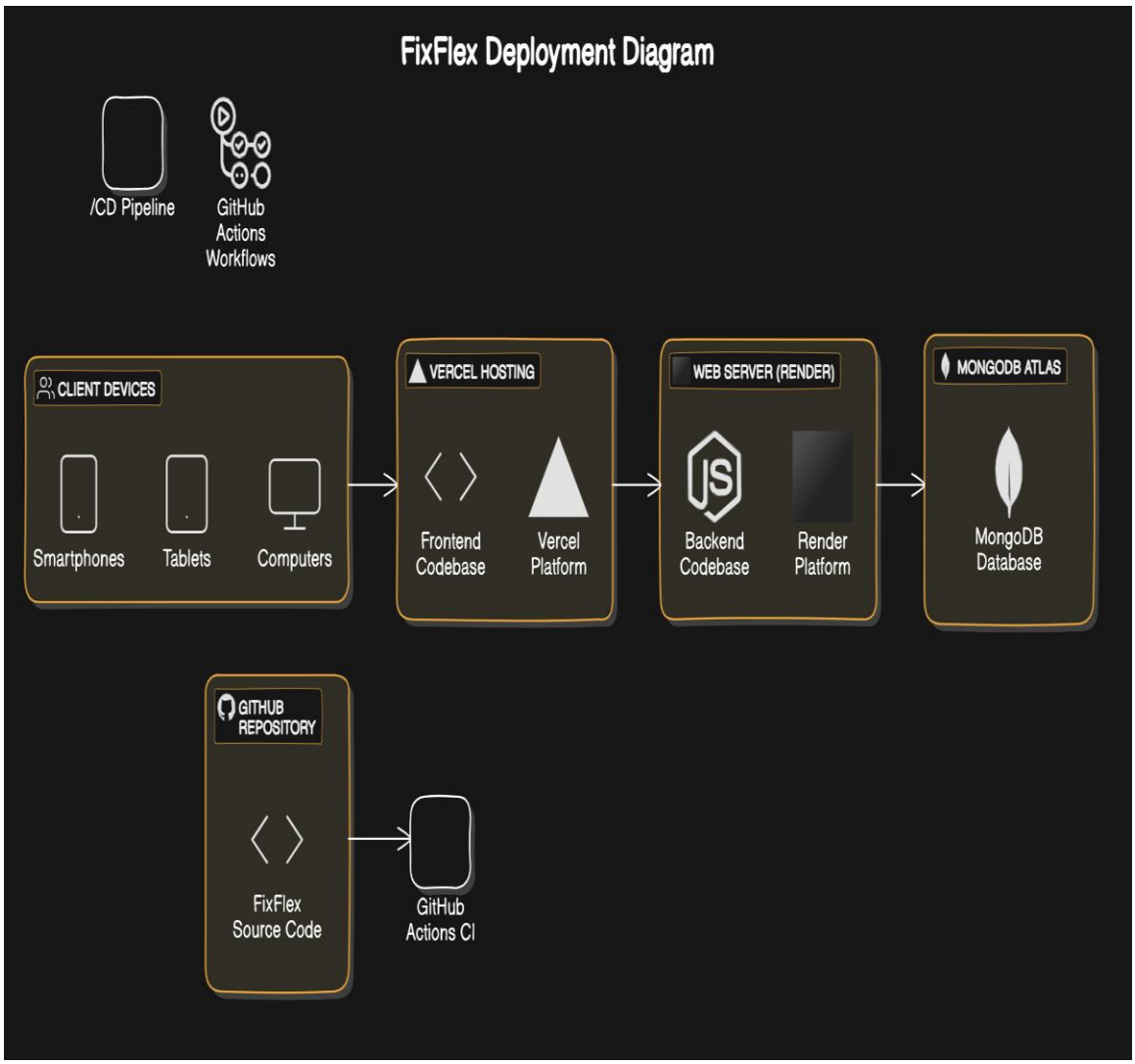

Tasker
User
FixFlex
Backend Server
Database
Enter FixFlex website
Login or Sign up
Verify credentials
Verify credentials
alt
[Logged in or Signed up successfully]
Redirect to dashboard
[Retry login or sign up]
Display error message
Create Task
Display post task page
Enter task details
Task details
Save task details
Make offer
alt
[accept offer]
select payment method
Payment
Paid
Save Paid
Save offer details
open chat
Chatting
Task Done
Display rate page
Rate Tasker
Send Rate
Save Rate
[reject offer]
send notification
Tasker
User
FixFlex
Backend Server
Database

Chapter 4

System Implementation and Results

4.1 Software Architecture:




FixFlex Deployment Diagram
/CD Pipeline
GitHub Actions Workflows
CLIENT DEVICES
Smartphones
Tablets
Computers
VERCEL HOSTING
Frontend Codebase
Vercel Platform
WEB SERVER (RENDER)
Backend Codebase
Render Platform
MONGODB ATLAS
MongoDB Database
GITHUB REPOSITORY
FixFlex Source Code
GitHub Actions CI

Backend:

The backend of FixFlex follows a modular, scalable, and maintainable architecture, leveraging TypeScript, Express.js, and MongoDB. The architecture is designed to ensure code reusability, separation of concerns, and scalability to accommodate future growth and changes in the application's requirements.

Model-View-Controller (MVC) Pattern

The backend architecture of FixFlex follows the Model-View-Controller (MVC) pattern, which divides the application into three main components:

- Models: Represents the data structure of the application, interacts with the database using Mongoose, and defines the schema for each resource.
- Controllers: Processes incoming requests, interacts with services, and returns responses to clients.
- Routes: Defines the API endpoints, maps HTTP methods to controller actions, and provides a standardized interface for clients to interact with the application.

Data Access Object (DAO) Pattern

The backend architecture of FixFlex employs the Data Access Object (DAO) pattern to separate the data access logic from the business logic. The DAO pattern encapsulates the database operations for each resource into a separate file, providing a clean and modular structure for managing data access.

Data Transfer Object (DTO) Pattern

The backend architecture of FixFlex uses Data Transfer Objects (DTOs) to transfer data between layers and components. DTOs define the structure of data exchanged between the client and server, ensuring consistency and type safety throughout the application.

Service Layer

The backend architecture of FixFlex includes a service layer that encapsulates the business logic of the application, interacts with the data access layer, and provides a centralized interface for controllers to access application functionality. The service layer helps maintain separation of concerns, improves code reusability, and facilitates testing and debugging.

Middleware

The backend architecture of FixFlex includes middleware functions that intercept incoming requests, perform actions, and pass control to the next middleware or route handler. Middleware functions are used for tasks such as authentication, error handling, logging, input validation, and more, providing a modular and extensible architecture for managing request processing.

Dependency Injection

The backend architecture of FixFlex employs dependency injection using the syringe package to manage component dependencies and improve code maintainability, flexibility, and testability. Dependency injection helps decouple components, promote code reuse, and facilitate the implementation of inversion of control (IoC) principles.

Database Schema

The database used for FixFlex is MongoDB, a NoSQL database that stores data in a flexible, JSON-like format.
See Database Schema file for more details.

The database schema for FixFlex consists of the following collections:

- Users: Stores user information such as name, email, phone number, password, role, profile picture, and verification status.
- Taskers: Stores tasker information such as user ID, description, skills, qualifications, availability, ratings, and earnings.
- Tasks: Stores task information such as user ID, title, description, category, location, status, budget, images, and timestamps.
- Categories: Stores category information such as name, description, parent category, and subcategories.
- Coupons: Stores coupon information such as code, discount amount, expiry date, and usage limit.
- Offers: Stores offer information such as task ID, tasker ID, amount, status, and timestamps.
- Reviews: Stores review information such as task ID, tasker ID, user ID, rating, comment, and timestamps.
- Chats: Stores chat information such as user IDs, task ID, tasker ID, and timestamps.

- Messages: Stores message information such as chat ID, user ID, tasker ID, content, and timestamps.
- Transactions: Stores transaction information such as user ID, tasker ID, amount, type, status, and timestamps.

RESTful API Design

The API design for FixFlex follows RESTful principles, with clear and predictable URL structures, HTTP methods, and status codes. The API endpoints are organized into resource-based routes, with each route corresponding to a specific resource or entity in the system. The API design includes the following key features:

- Resource-Based Routes: Organizes API endpoints into resource-based routes such as /users, /tasks, /taskers, /offers, /reviews, /categories, and /coupons.
- CRUD Operations: Implements Create, Read, Update, and Delete operations for all resources using HTTP methods such as GET, POST, PUT, PATCH, and DELETE.
- Pagination and Filtering: Supports pagination and filtering for listing resources, allowing users to limit the number of results and filter based on specific criteria.
- Search and Sorting: Allows users to search for resources based on keywords, categories, locations, and other criteria, enhancing resource discovery.
- Error Handling: Provides informative error messages, status codes, and error responses to help clients understand and handle errors effectively.

Modular Structure

- **Controllers**: Handle incoming requests, process data, and send responses to clients.
- **Services**: Contain business logic, interact with the database, and perform operations on data.
- **Models**: Define data structures and schemas for MongoDB collections using Mongoose.
- **Routes**: Define API endpoints, route requests to controllers, and handle HTTP methods.
- **Middleware**: Implement custom middleware functions for authentication, validation, error handling, etc.
- **Exceptions**: Define custom exception classes to handle errors and exceptions in the application.
- **Helpers**: Contain utility functions, helper classes, and third-party integrations to assist with common tasks.
- **Config**: Store configuration settings, environment variables, and validation rules.

- **DB**: Contain data access objects (DAOs) to interact with the MongoDB database.
- **Docs**: Define Swagger documentation files for API endpoints and models.
- **DTOs**: Define data transfer objects (DTOs) to transfer data between layers and components.
- **Interfaces**: Define TypeScript interfaces for data structures, models, and API requests and responses.
- **Middleware**: Implement custom middleware functions for authentication, validation, error handling, etc.
- **Routes**: Define API endpoints, route requests to controllers, and handle HTTP methods.
- **Services**: Contain business logic, interact with the database, and perform operations on data.
- **Sockets**: Implement real-time communication between clients and servers using Web Sockets.
- **Types**: Define custom TypeScript types, interfaces, and declarations for better type safety and code quality.

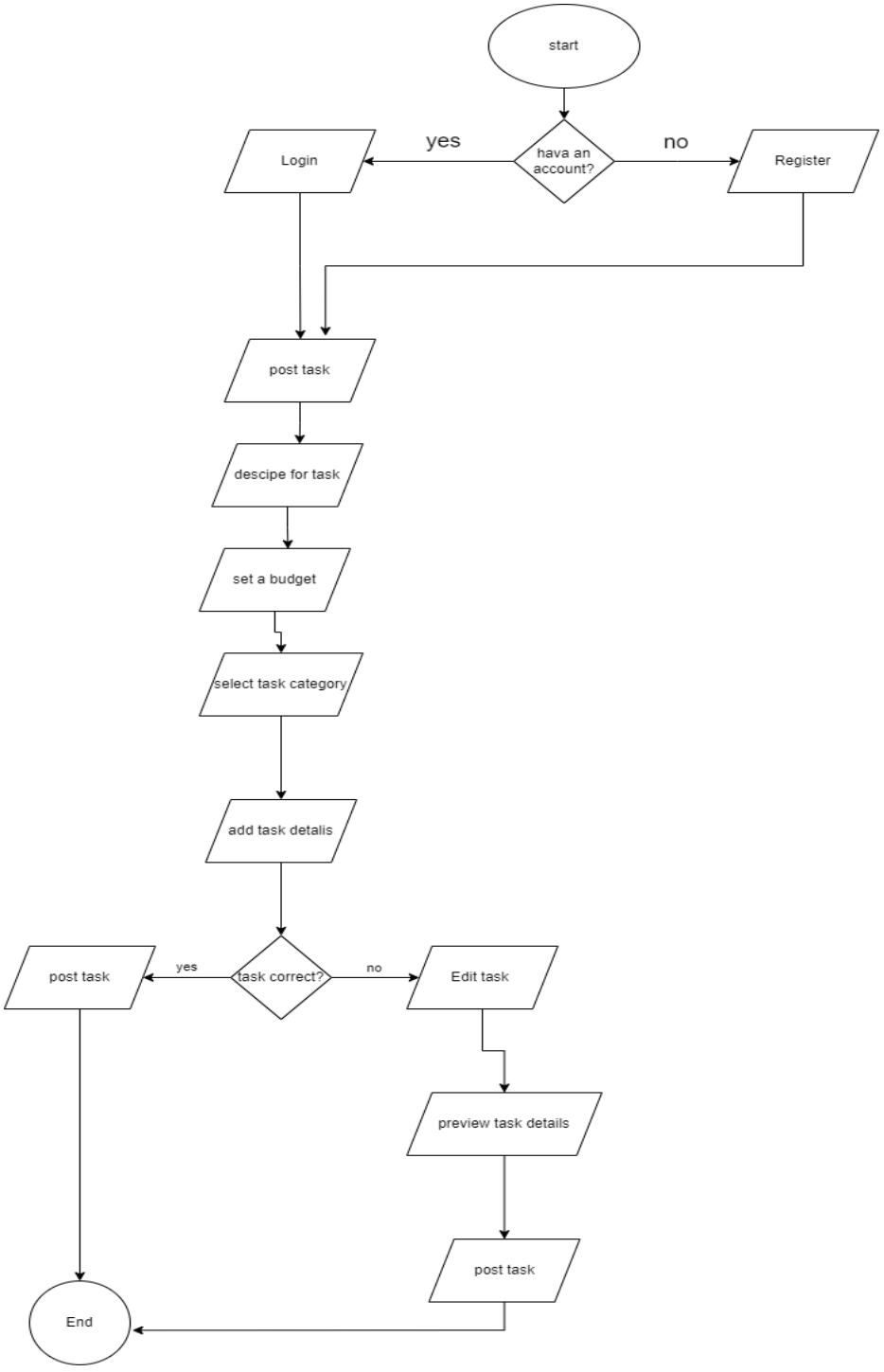
Error Handling

The backend architecture of FixFlex includes a robust error handling mechanism to catch and handle exceptions gracefully, providing informative error messages, status codes, and error responses to clients. The error handling mechanism includes the following key features:

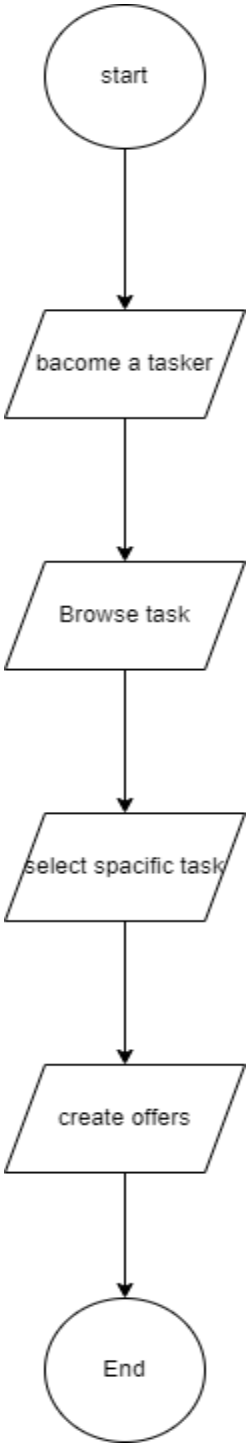
- Custom Exceptions: Defines custom exception classes to handle errors and exceptions in the application, providing detailed error messages and status codes.
- Error Middleware: Implements error middleware functions to catch and handle exceptions, log errors, and send error responses to clients.
- Global Error Handling: Centralizes error handling logic in a global error handler middleware to ensure consistent error responses across the application.
- Error Logging: Logs errors and exceptions to track application behavior, monitor performance, and debug issues effectively.
- Error Response Format: Standardizes error responses with a consistent format, including status codes, error messages, error details, and stack traces.

4.2 Flowchart:

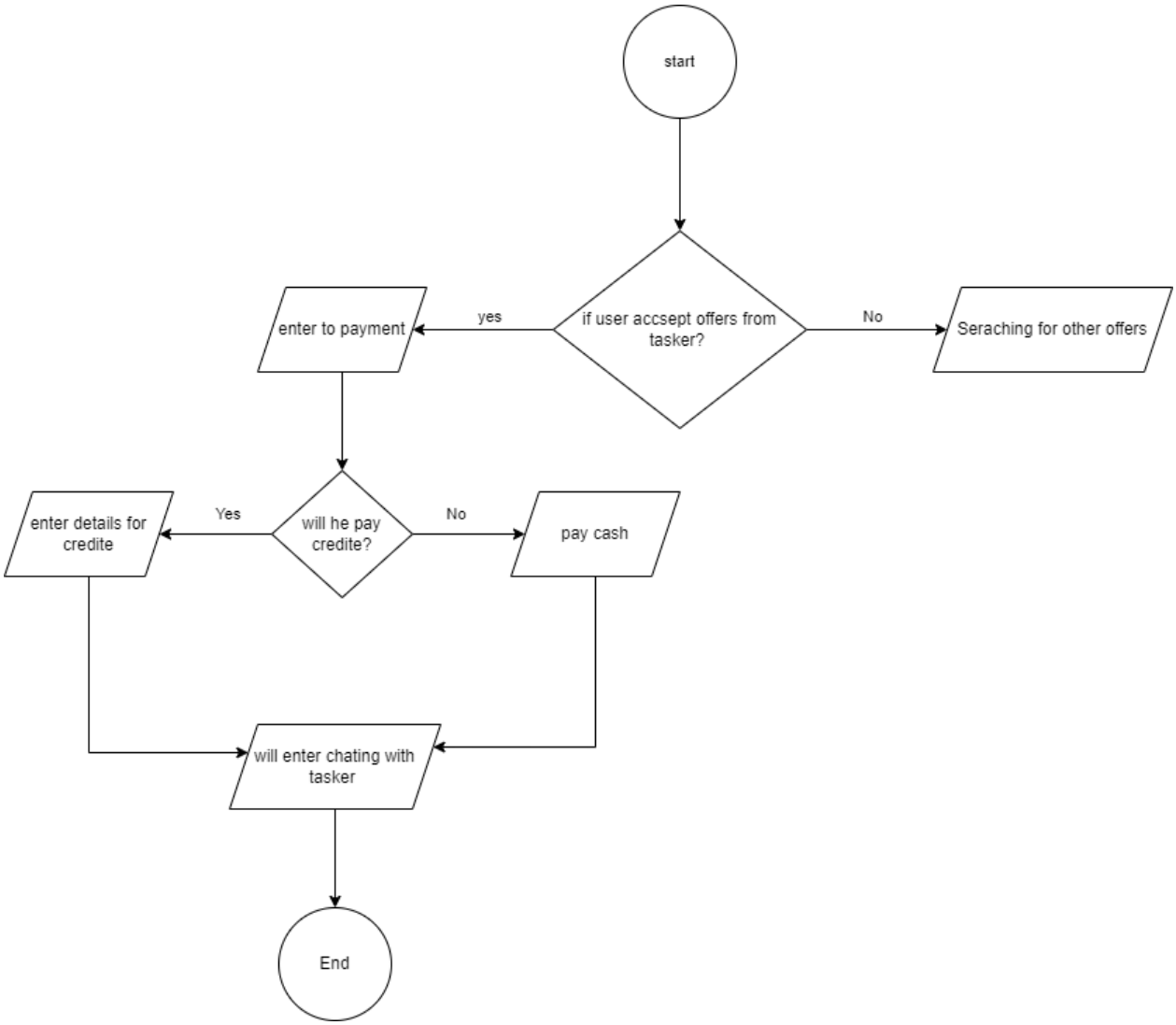
-post task



- **Tasker make offers**



- **Accept Offers**



4.3 AI Integration with Gemini API

In this section, we describe how AI, specifically through the use of the Gemini API, is integrated into our project to enhance task management functionalities.

4.3.1 Task Categorization

Using the Gemini API, we predict the appropriate category for each task based on its title. This categorization helps in organizing tasks more effectively and ensures they are assigned to the correct departments or personnel.

4.3.2 Task Mode Prediction

The Gemini API also predicts whether a task should be performed online or in person (offline). This is crucial for planning and resource allocation, especially in projects that have both remote and on-site components.

4.3.3 Budget Estimation

Another feature of our AI integration is the estimation of the budget required for each task. The Gemini API analyzes the task title and other relevant details to provide a budget estimation, helping in financial planning and cost management.

4.3.4 Detailed Task Description

The Gemini API enhances the task details by providing additional context and information based on the initial task title. This helps in creating comprehensive task descriptions that are clear and detailed.

4.3.5 Accuracy

Our AI system, powered by the Gemini API, has been tested extensively and has demonstrated an accuracy of 98% across various tasks and predictions. This high level of accuracy ensures reliable and efficient task management.

Chapter 5

Testing

5.1 Test Case

Testing the LOGIN function:

TEST TITLE	PRIORITY	TEST CASE ID	TEST NUMBER	TEST DATE
Login	High	202000	1	4-6-2024

TEST DESCRIPTION	TEST DESIGNED BY	TEST EXECUTED BY	EXECUTION DATE
Testing the login functionality	Basmala Yasser	Esraa Raafat	5-6-2024

No.	Test Scenarios	Expected Results	Actual Results	Negative/ Positive Test
1	Verify if a user will be able to login with a valid email and valid password.	User login	login succeeded	Positive
2	Verify if a user cannot login with a valid email and an invalid password.	user cannot login	user cannot login	Positive
3	Verify the 'Forgot Password' functionality.	reset password then user login	User login	Positive
4	Verify the 'login with google' functionality.	User can login with google account	Login succeeded	Positive
5	Verify the messages for invalid login.	A message appears in case of incorrect user login	The message appears to the user	Positive
6	Verify the login button is work	No error in button work	Login succeeded	Positive

Testing the POST TASK function:

TEST TITLE	PRIORITY	TEST CASE ID	TEST NUMBER	TEST DATE
Post task	High	202200	1	4-6-2024

TEST DESCRIPTION	TEST DESIGNED BY	TEST EXECUTED BY	EXECUTION DATE
Ensure that a user can post a task with details, budget, and time.	Basmala Yasser	Esraa Raafat	5-6-2024

- **Preconditions**: User is logged in.
- **Steps**:

1. Navigate to the "Post a Task" page.
2. Enter task details (title, description, category, location).
3. Enter the budget for the task.
4. Enter the time for task completion.
5. Click on the "Post" button.

- **Expected Result**: The task should be posted successfully and should be visible to task runners in the task listings.

Testing the BROWSE TASK function:

TEST TITLE	PRIORITY	TEST CASE ID	TEST NUMBER	TEST DATE
Browse task	High	202200	1	4-6-2024

TEST DESCRIPTION	TEST DESIGNED BY	TEST EXECUTED BY	EXECUTION DATE
Ensure that a tasker can browse through available tasks.	Basmala Yasser	Esraa Raafat	5-6-2024

- **Preconditions**: User or tasker is logged in.
- **Steps**:

1. Navigate to the "Browse Tasks" page.
2. View the list of available tasks.
3. Filter tasks by category.
4. Click on a task to view its details.

- **Expected Result**: The tasker should be able to browse the tasks, apply filters, and view detailed information for each task.

5.2 Bug Report

Bug report

Bug Title: Complete Task Button Not Functioning

Bug Description: The "Complete Task" button does not respond when clicked, preventing users from marking tasks as complete.

Severity: High

Priority: High

Bug ID:232

Environment:

- **Platform**: Web
- **Browser**: Chrome v91.0
- **Operating System**: Windows 10
- **Application Version**: VS (1.3.0)

Assigned to: Menna Alaa Eldin

Reported By:

- [Basmala Yasser]
- Date: [5-4-2024]

Steps to Reproduce

1. Log in to the system as a user.
2. Navigate to the task list.
3. Select a task that is in progress.
4. Click on the "Complete Task" button.

Expected Result

- The task should be marked as complete.
- The task status should be updated in the database.
- The task should appear in the list of completed tasks.

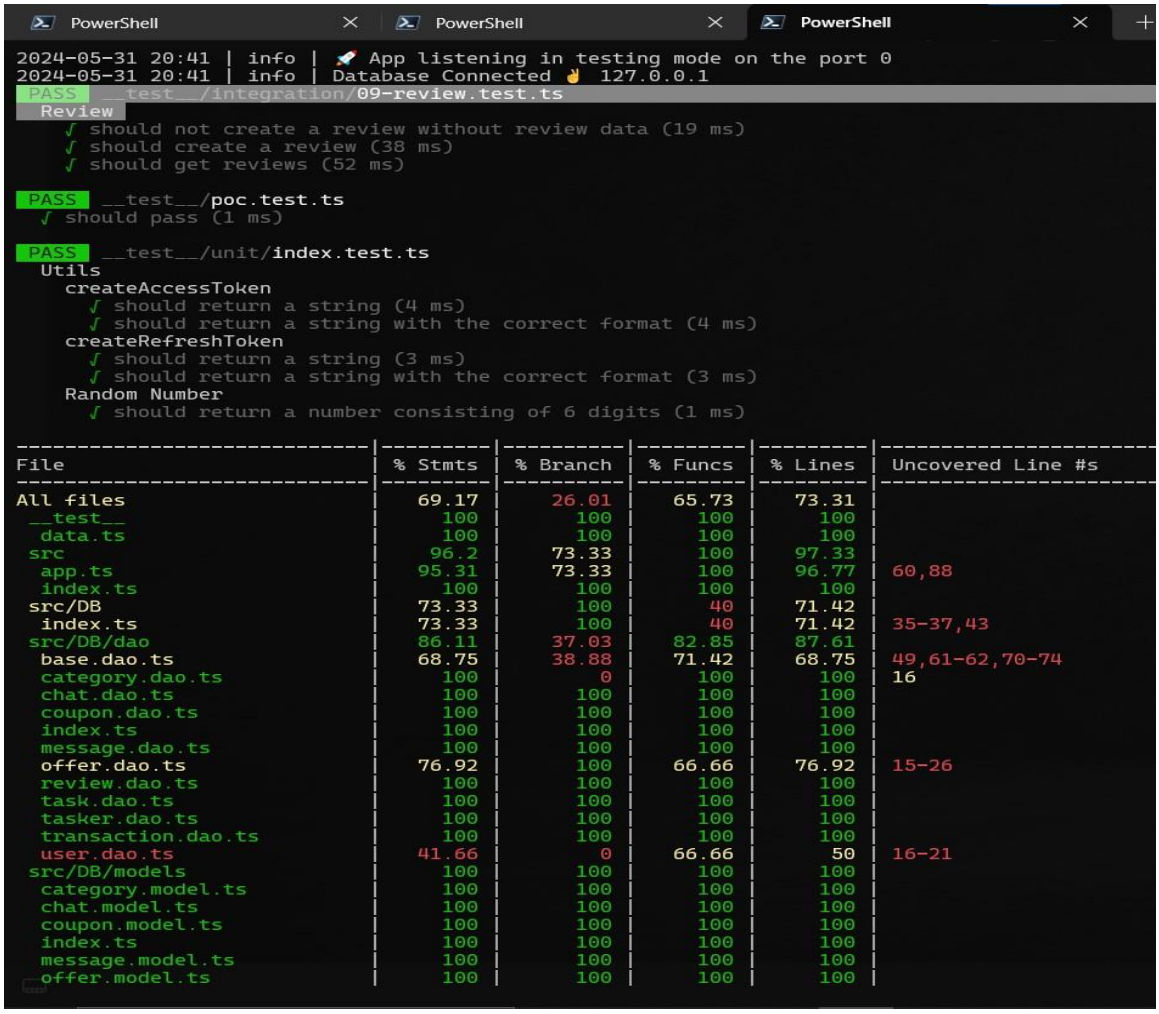
Actual Result

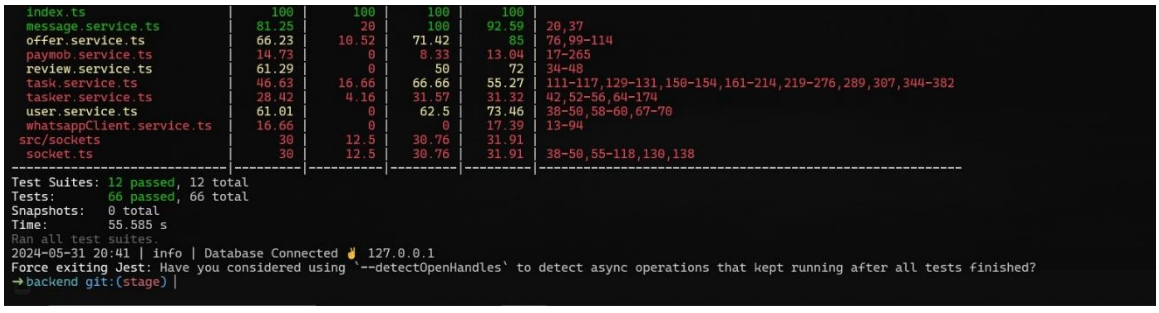
- The "Complete Task" button does not respond when clicked.
- The task status remains unchanged.

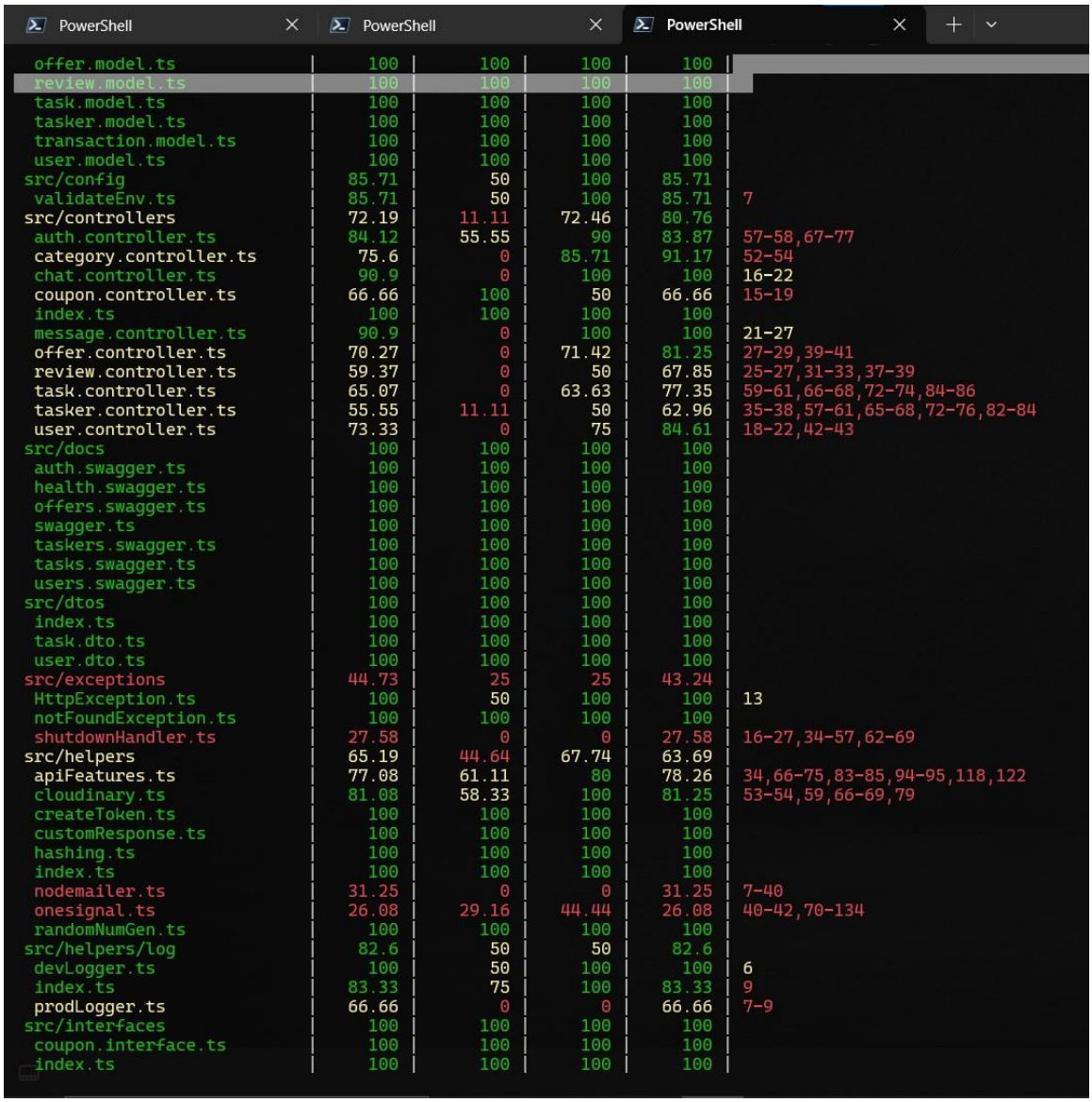
5.3 Test System


PowerShell
PowerShell
PowerShell
→ backend git:(stage) npm run test:coverage
> fixflex@1.0.0 test:coverage
> cross-env NODE_ENV=testing jest --runInBand --verbose --coverage
2024-05-31 20:40 | info | 🚀 App listening in testing mode on the port 0
2024-05-31 20:40 | info | Database Connected ✌ 127.0.0.1
PASS __test__/app.test.ts (7.773 s)
app
GET /api/v1
given the endpoint exist
√ should return a 200 status with with a json message (179 ms)
GET /api/v1/healthz
given the endpoint exist
√ should return a 200 status with with a json message (55 ms)
GET /not-found-endpoint
given the endpoint does not exist
√ should return a 404 status with not found message (28 ms)
2024-05-31 20:40 | info | 🚀 App listening in testing mode on the port 0
2024-05-31 20:40 | info | Database Connected ✌ 127.0.0.1
PASS __test__/integration/00-auth.test.ts
Authentication
POST /api/v1/auth/signup
√ should return 201 and create new user (403 ms)
√ should return 409 E-Mail address is already exists (24 ms)
√ fails signup without required fields (14 ms)
POST /api/v1/auth/login
√ should login with correct credentials (211 ms)
√ should return 401 Incorrect email or password` (200 ms)
√ fails login without required fields (14 ms)
GET /api/v1/auth/refresh-token
√ should return 401 Unauthorized (14 ms)
√ should return 200 and refresh token (227 ms)
GET /api/v1/auth/logout
√ should return 200 and logout (222 ms)
GET /api/v1/auth/forgot-password
√ should return 200 and send reset code (41 ms)
POST /api/v1/auth/verify-reset-code
√ should return 200 and verify reset code (21 ms)
PATCH /api/v1/auth/reset-password
√ should return 200 and reset password (207 ms)
PATCH /api/v1/auth/change-password
√ should return 200 and change password (401 ms)
2024-05-31 20:40 | info | 🚀 App listening in testing mode on the port 0
2024-05-31 20:40 | info | Database Connected ✌ 127.0.0.1
PASS __test__/integration/01-user.test.ts (5.95 s)
User
GET /api/v1/users/me
√ should return 200 and user data (21 ms)
√ should return 401 if no token is provided (14 ms)
PATCH /api/v1/users/me


PowerShell
PowerShell
PowerShell
PASS __test__/integration/05-task.test.ts (12.632 s)
Task
√ should create a task (2342 ms)
√ should get all tasks (26 ms)
√ should get a task by id (44 ms)
√ should update a task (2968 ms)
√ should upload a task images (4131 ms)
2024-05-31 20:40 | info | 🚀 App listening in testing mode on the port 0
2024-05-31 20:40 | info | Database Connected ✌ 127.0.0.1
console.log
{ id: '', errors: ['All included players are not subscribed'] } { id: '', errors: ['A
at TaskService.log [as completeTask] (src/services/task.service.ts:338:13)
PASS __test__/integration/06-offer.test.ts
Offer
√ should not create an offer without offer data (21 ms)
√ should create an offer (57 ms)
√ should update an offer (25 ms)
√ should get offers (17 ms)
√ should accept the offer and pay cash (415 ms)
Complete Task
√ should complete the task (433 ms)
2024-05-31 20:41 | info | 🚀 App listening in testing mode on the port 0
2024-05-31 20:41 | info | Database Connected ✌ 127.0.0.1
console.log
665a0b8600ae2667cc1ab23c
at ChatService.log [as createChat] (src/services/chat.service.ts:31:13)
PASS __test__/integration/07-chat.test.ts
Chat
√ should not create a chat room without tasker id (15 ms)
√ should create a chat room (44 ms)
√ should get user chats (20 ms)
√ should get chat by id (34 ms)
2024-05-31 20:41 | info | 🚀 App listening in testing mode on the port 0
2024-05-31 20:41 | info | Database Connected ✌ 127.0.0.1
PASS __test__/integration/08-message.test.ts
Message
√ should not create a message without message data (18 ms)
√ should create a message (32 ms)
√ should get messages by chat id (19 ms)
√ should delete a message (42 ms)
2024-05-31 20:41 | info | 🚀 App listening in testing mode on the port 0
2024-05-31 20:41 | info | Database Connected ✌ 127.0.0.1
PASS __test__/integration/09-review.test.ts
Review


PowerShell
PowerShell
PowerShell
2024-05-31 20:41 | info | 🚀 App listening in testing mode on the port 0
2024-05-31 20:41 | info | Database Connected ✌ 127.0.0.1
PASS __test__/integration/09-review.test.ts
Review
✓ should not create a review without review data (19 ms)
✓ should create a review (38 ms)
✓ should get reviews (52 ms)
PASS __test__/poc.test.ts
✓ should pass (1 ms)
PASS __test__/unit/index.test.ts
Utils
createAccessToken
✓ should return a string (4 ms)
✓ should return a string with the correct format (4 ms)
createRefreshToken
✓ should return a string (3 ms)
✓ should return a string with the correct format (3 ms)
Random Number
✓ should return a number consisting of 6 digits (1 ms)
File | % Stmts | % Branch | % Funcs | % Lines | Uncovered Line #s
All files | 69.17 | 26.01 | 65.73 | 73.31 |
__test__ | 100 | 100 | 100 | 100 |
data.ts | 100 | 100 | 100 | 100 |
src | 96.2 | 73.33 | 100 | 97.33 |
app.ts | 95.31 | 73.33 | 100 | 96.77 | 60,88
index.ts | 100 | 100 | 100 | 100 |
src/DB | 73.33 | 100 | 40 | 71.42 |
index.ts | 73.33 | 100 | 40 | 71.42 | 35-37,43
src/DB/dao | 86.11 | 37.03 | 82.85 | 87.61 |
base.dao.ts | 68.75 | 38.88 | 71.42 | 68.75 | 49,61-62,70-74
category.dao.ts | 100 | 0 | 100 | 100 | 16
chat.dao.ts | 100 | 100 | 100 | 100 |
coupon.dao.ts | 100 | 100 | 100 | 100 |
index.ts | 100 | 100 | 100 | 100 |
message.dao.ts | 100 | 100 | 100 | 100 |
offer.dao.ts | 76.92 | 100 | 66.66 | 76.92 | 15-26
review.dao.ts | 100 | 100 | 100 | 100 |
task.dao.ts | 100 | 100 | 100 | 100 |
tasker.dao.ts | 100 | 100 | 100 | 100 |
transaction.dao.ts | 100 | 100 | 100 | 100 |
user.dao.ts | 41.66 | 0 | 66.66 | 50 | 16-21
src/DB/models | 100 | 100 | 100 | 100 |
category.model.ts | 100 | 100 | 100 | 100 |
chat.model.ts | 100 | 100 | 100 | 100 |
coupon.model.ts | 100 | 100 | 100 | 100 |
index.ts | 100 | 100 | 100 | 100 |
message.model.ts | 100 | 100 | 100 | 100 |
offer.model.ts | 100 | 100 | 100 | 100 |


index.ts | 100 | 100 | 100 | 100 |
message.service.ts | 81.25 | 20 | 100 | 92.59 | 20,37
offer.service.ts | 66.23 | 10.52 | 71.42 | 85 | 76,99-114
paymob.service.ts | 14.73 | 0 | 8.33 | 13.04 | 17-265
review.service.ts | 61.29 | 0 | 50 | 72 | 34-48
task.service.ts | 46.63 | 16.66 | 66.66 | 55.27 | 111-117,129-131,150-154,161-214,219-276,289,307,344-382
tasker.service.ts | 28.42 | 4.16 | 31.57 | 31.32 | 42,52-56,64-174
user.service.ts | 61.01 | 0 | 62.5 | 73.46 | 38-50,58-60,67-70
whatsappClient.service.ts | 16.66 | 0 | 0 | 17.39 | 13-94
src/sockets | 30 | 12.5 | 30.76 | 31.91 |
socket.ts | 30 | 12.5 | 30.76 | 31.91 | 38-50,55-118,130,138
Test Suites: 12 passed, 12 total
Tests: 66 passed, 66 total
Snapshots: 0 total
Time: 55.585 s
Ran all test suites.
2024-05-31 20:41 | info | Database Connected ✌ 127.0.0.1
Force exiting Jest: Have you considered using `--detectOpenHandles` to detect async operations that kept running after all tests finished?
→backend git:(stage) |



offer.model.ts	100	100	100	100	
review.model.ts	100	100	100	100	
task.model.ts	100	100	100	100	
tasker.model.ts	100	100	100	100	
transaction.model.ts	100	100	100	100	
user.model.ts	100	100	100	100	
src/config	85.71	50	100	85.71	
validateEnv.ts	85.71	50	100	85.71	7
src/controllers	72.19	11.11	72.46	80.76	
auth.controller.ts	84.12	55.55	90	83.87	57-58,67-77
category.controller.ts	75.6	0	85.71	91.17	52-54
chat.controller.ts	90.9	0	100	100	16-22
coupon.controller.ts	66.66	100	50	66.66	15-19
index.ts	100	100	100	100	
message.controller.ts	90.9	0	100	100	21-27
offer.controller.ts	70.27	0	71.42	81.25	27-29,39-41
review.controller.ts	59.37	0	50	67.85	25-27,31-33,37-39
task.controller.ts	65.07	0	63.63	77.35	59-61,66-68,72-74,84-86
tasker.controller.ts	55.55	11.11	50	62.96	35-38,57-61,65-68,72-76,82-84
user.controller.ts	73.33	0	75	84.61	18-22,42-43
src/docs	100	100	100	100	
auth.swagger.ts	100	100	100	100	
health.swagger.ts	100	100	100	100	
offers.swagger.ts	100	100	100	100	
swagger.ts	100	100	100	100	
taskers.swagger.ts	100	100	100	100	
tasks.swagger.ts	100	100	100	100	
users.swagger.ts	100	100	100	100	
src/dtos	100	100	100	100	
index.ts	100	100	100	100	
task.dto.ts	100	100	100	100	
user.dto.ts	100	100	100	100	
src/exceptions	44.73	25	25	43.24	
HttpException.ts	100	50	100	100	13
notFoundException.ts	100	100	100	100	
shutdownHandler.ts	27.58	0	0	27.58	16-27,34-57,62-69
src/helpers	65.19	44.64	67.74	63.69	
apiFeatures.ts	77.08	61.11	80	78.26	34,66-75,83-85,94-95,118,122
cloudinary.ts	81.08	58.33	100	81.25	53-54,59,66-69,79
createToken.ts	100	100	100	100	
customResponse.ts	100	100	100	100	
hashing.ts	100	100	100	100	
index.ts	100	100	100	100	
nodemailer.ts	31.25	0	0	31.25	7-40
onesignal.ts	26.08	29.16	44.44	26.08	40-42,70-134
randomNumGen.ts	100	100	100	100	
src/helpers/log	82.6	50	50	82.6	
devLogger.ts	100	50	100	100	6
index.ts	83.33	75	100	83.33	9
prodLogger.ts	66.66	0	0	66.66	7-9
src/interfaces	100	100	100	100	
coupon.interface.ts	100	100	100	100	
index.ts	100	100	100	100	

coupon.interface.ts	100	100	100	100	
index.ts	100	100	100	100	
offer.interface.ts	100	100	100	100	
task.interface.ts	100	100	100	100	
transaction.interface.ts	100	100	100	100	
user.interface.ts	100	100	100	100	
src/middleware	88.88	50	100	87.83	
auth.middleware.ts	85.71	53.84	100	85.36	23,32,50,56,60,71
index.ts	100	100	100	100	
middleware.ts	90.47	33.33	100	89.47	52,63
uploadImages.middleware.ts	93.33	50	100	90.9	12
src/middleware/errors	53.62	28.57	33.33	58.06	
error.middleware.ts	43.85	25	20	48	13,16,19,22,36-37,41-47,51-53,61-71,77,102-105
index.ts	100	100	100	100	
validation.middleware.ts	100	100	100	100	
src/middleware/validation	59.2	18.91	46.66	61.01	
auth.validator.ts	100	100	100	100	
category.validator.ts	100	100	100	100	
chat.validator.ts	100	100	100	100	
coupon.validator.ts	40	0	0	40	14-28
index.ts	100	100	100	100	
isMongoID.validator.ts	100	100	100	100	
message.validator.ts	100	100	100	100	
offer.validator.ts	80	0	100	80	17,43
review.validator.ts	100	100	100	100	
tasker.validator.ts	53.33	25	50	53.33	21,24,49-56
tasks.validator.ts	28	15.25	37.5	29.54	16,19,45-70,96-113,122-140
user.validator.ts	100	100	100	100	
src/routes	98.46	25	93.1	98.44	
auth.route.ts	100	100	100	100	
category.route.ts	100	100	100	100	
chat.route.ts	100	100	100	100	
coupon.route.ts	100	100	100	100	
healthz.route.ts	100	100	100	100	
index.ts	100	100	100	100	
message.route.ts	100	100	100	100	
offer.route.ts	100	100	100	100	
review.route.ts	100	100	100	100	
routes.ts	100	100	100	100	
task.route.ts	100	100	100	100	
tasker.route.ts	100	100	100	100	
user.route.ts	100	100	100	100	
webhooks.route.ts	76.47	0	50	76.47	21-25,29
src/services	50.55	12.9	51.45	56.37	
auth.service.ts	70.83	35	88.88	72.82	44,59-85,96,100,113,135-140,155,168,172
category.service.ts	80	28.57	85.71	88.23	17,62-64
chat.service.ts	90.47	0	100	100	23-26
coupon.service.ts	85.71	100	50	85.71	11
index.ts	100	100	100	100	
message.service.ts	81.25	20	100	92.59	20,37
offer.service.ts	66.23	10.52	71.42	85	76,99-114
paymob.service.ts	14.73	0	8.33	13.04	17-265
review.service.ts	61.29	0	50	72	34-48

All files

69.17% Statements 1692/2446 **26.01%** Branches 128/492 **65.73%** Functions 234/356 **73.31%** Lines 1651/2252

Press *n* or *j* to go to the next uncovered block, *b*, *p* or *k* for the previous block.

Filter:

File		Statements		Branches		Functions		Lines	
__test__		100%	7/7	100%	0/0	100%	0/0	100%	7/7
src		96.2%	76/79	73.33%	11/15	100%	12/12	97.33%	73/75
src/DB		73.33%	11/15	100%	0/0	40%	2/5	71.42%	10/14
src/DB/dao		86.11%	93/108	37.03%	10/27	82.85%	29/35	87.61%	92/105
src/DB/models		100%	71/71	100%	1/1	100%	2/2	100%	71/71
src/config		85.71%	6/7	50%	1/2	100%	0/0	85.71%	6/7
src/controllers		72.19%	296/410	11.11%	6/54	72.46%	50/69	80.76%	294/364
src/docs		100%	41/41	100%	0/0	100%	0/0	100%	41/41
src/dtos		100%	29/29	100%	0/0	100%	2/2	100%	29/29
src/exceptions		44.73%	17/38	25%	1/4	25%	2/8	43.24%	16/37
src/helpers		65.19%	118/181	44.64%	25/56	67.74%	21/31	63.69%	107/168
src/helpers/log		82.6%	19/23	50%	4/8	50%	1/2	82.6%	19/23
src/interfaces		100%	47/47	100%	14/14	100%	7/7	100%	47/47
src/middleware		88.88%	72/81	50%	9/18	100%	11/11	87.83%	65/74
src/middleware/errors		53.62%	37/69	28.57%	6/21	33.33%	4/12	58.06%	36/62
src/middleware/validation		59.2%	74/125	18.91%	14/74	46.66%	7/15	61.01%	72/118
src/routes		98.46%	256/260	25%	1/4	93.1%	27/29	98.44%	253/257
src/services		50.55%	407/805	12.9%	24/186	51.45%	53/103	56.37%	398/706
src/sockets		30%	15/50	12.5%	1/8	30.76%	4/13	31.91%	15/47

GTmetrix The web should be fast.

Executive Summary

Performance Report for:
https://www.fixflex.tech/

Report generated: Wed, Jun 5, 2024 4:43 PM -0700
Test Server Location: Vancouver, Canada
Using: Chrome 117.0.0.0, Lighthouse 11.0.0

B	Performance	Structure	L. Contentful Paint	T. Blocking Time	C. Layout Shift
	92%	81%	1.8s	72ms	0

Top Issues

High	Avoid enormous network payloads LCP	Total size was 33.9MB
Med-High	Properly size images	Potential savings of 32.8MB
Med	Serve static assets with an efficient cache policy	Potential savings of 6.99MB
Low	Serve images in next-gen formats	Potential savings of 9.30MB
Low	Preload Largest Contentful Paint image LCP	91ms

Focus on these audits first

These audits likely have the largest impact on your page performance.

Structure audits do not directly affect your Performance Score, but improving the audits seen here can help as a starting point for overall performance gains.

Page Details

1.8s
Fully Loaded Time

Total Page Size - 33.9MB

Total Page Requests - 40

HTML JS CSS IMG Video Font Other

How does this affect me?

Modern web users have a short attention span and expect a fast and seamless website experience. Delivering that fast experience can result in more traffic, more conversions, and more happiness.

As if you didn't need more incentive, **Google use Page Speed and Page Experience (including Web Vitals) signals in their ranking algorithm.**

About GTmetrix

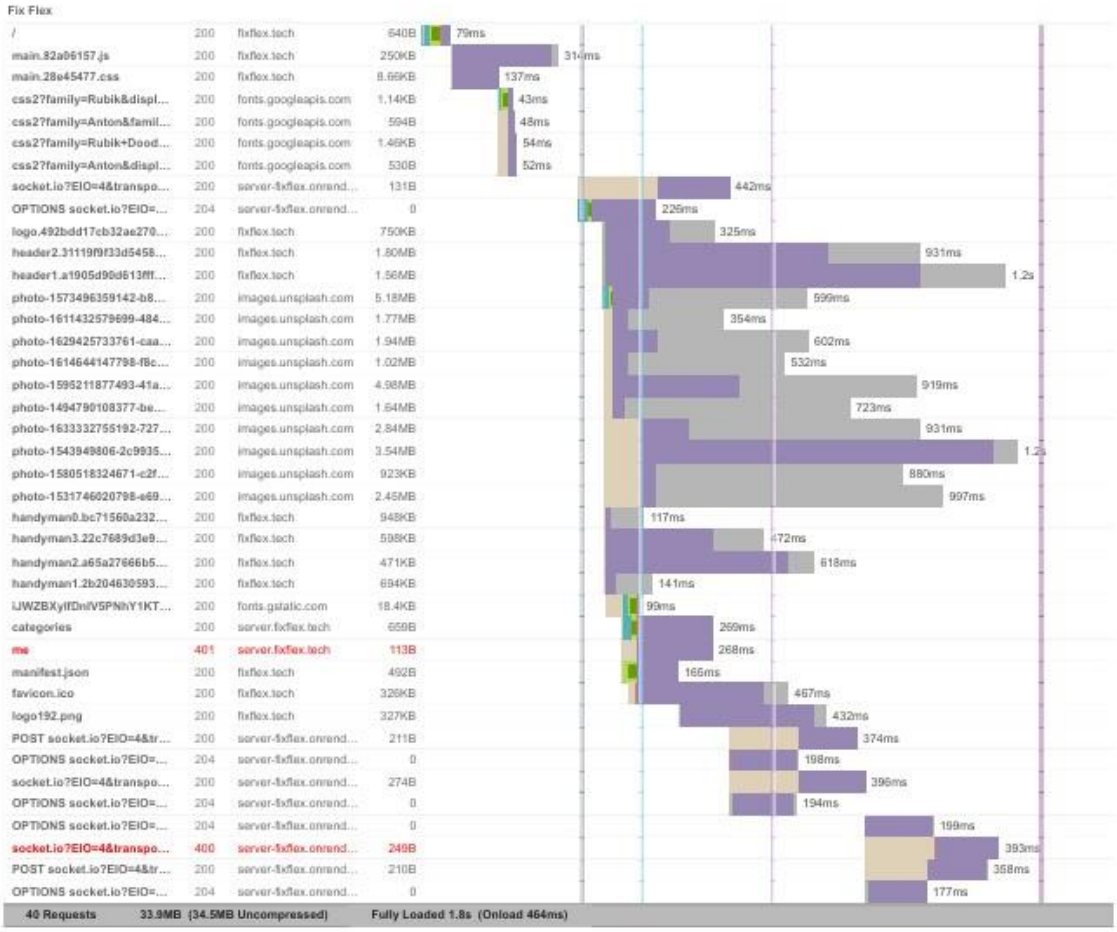
GTmetrix was developed as a tool for customers to easily test the performance of their webpages.

Learn more about us.

GTmetrix

Waterfall Chart

The waterfall chart displays the loading behaviour of your site in your selected browser. It can be used to discover simple issues such as 404's or more complex issues such as external resources blocking page rendering.



GTmetrix

Performance

Performance Metrics

First Contentful Paint

Good - Nothing to do here

How quickly content like text or images are painted onto your page. A good user experience is 0.9s or less.

636ms

Time to Interactive

Good - Nothing to do here

How long it takes for your page to become fully interactive. A good user experience is 2.5s or less.

1.0s

Speed Index

Good - Nothing to do here

How quickly the contents of your page are visibly populated. A good user experience is 1.3s or less.

836ms

Total Blocking Time

Good - Nothing to do here

How much time is blocked by scripts during your page loading process. A good user experience is 150ms or less.

72ms

Largest Contentful Paint

Longer than recommended

How long it takes for the largest element of content (i.e., a hero image) to be painted on your page. A good user experience is 1.2s or less.

1.8s

Cumulative Layout Shift

Good - Nothing to do here

How much your page's layout shifts as it loads. A good user experience is a score of 0.1 or less.

0

Browser Timings

Redirect	0ms	Connect	53ms	Backend	26ms
TTFB	79ms	DOM Int.	92ms	DOM Loaded	464ms
Onload	464ms	First Paint	636ms	Fully Loaded	1.8s

GTmetrix

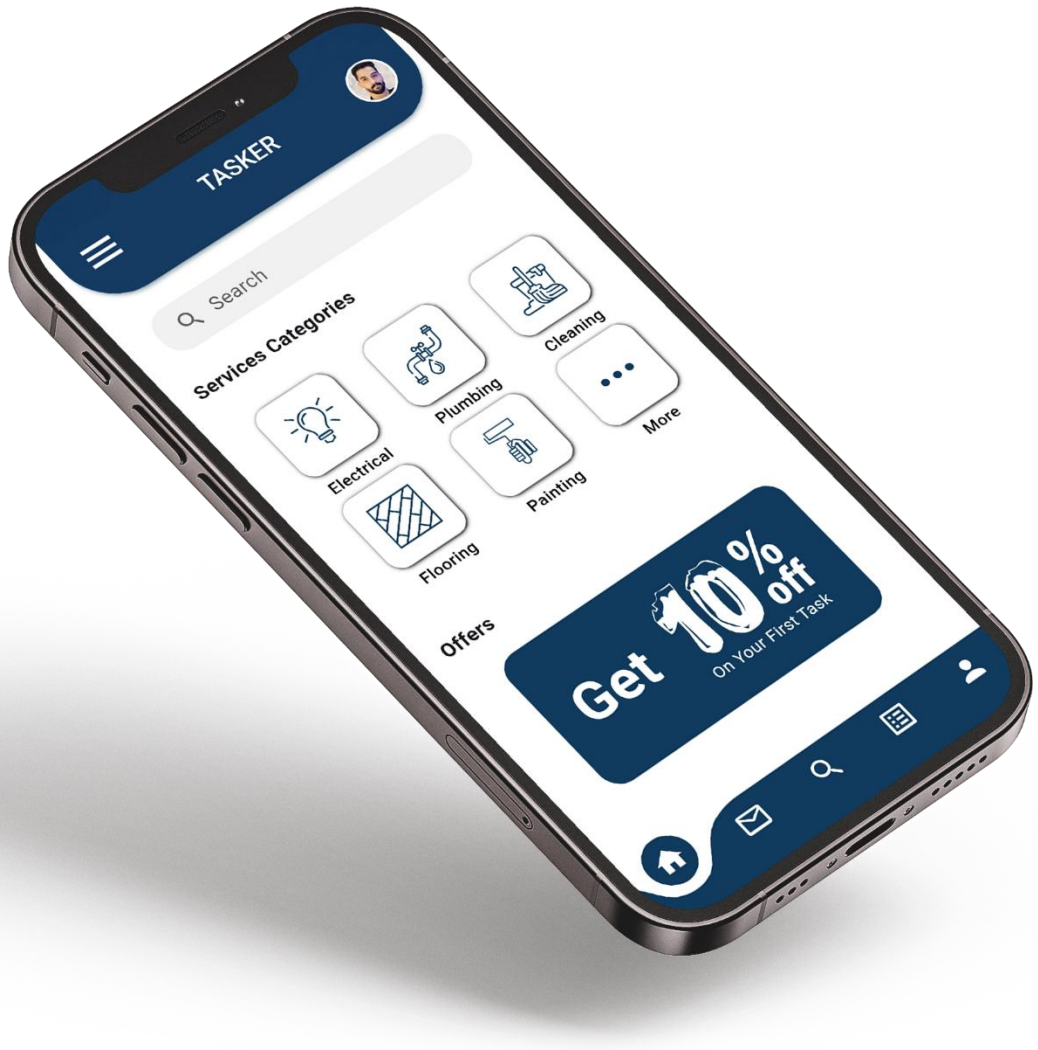
Structure Audits

IMPACT	AUDIT	
High	Avoid enormous network payloads LCP	Total size was 33.9MB
Med-High	Properly size images	Potential savings of 32.8MB
Med	Serve static assets with an efficient cache policy	Potential savings of 6.99MB
Low	Serve images in next-gen formats	Potential savings of 9.30MB
Low	Preload Largest Contentful Paint image LCP	91ms
Low	Avoid an excessive DOM size TBT	402 elements
Low	Efficiently encode images	Potential savings of 528KB
Low	Avoid long main-thread tasks TBT	2 long tasks found
Low	Reduce JavaScript execution time TBT	167ms spent executing JavaScript
Low	Avoid non-composited animations CLS	1 animated element found
Low	Avoid chaining critical requests FCP LCP	4 chains found
Low	Reduce unused JavaScript LCP	Potential savings of 134KB
N/A	Largest Contentful Paint element LCP	1,820 ms
N/A	Reduce initial server response time FCP LCP	Root document took 25ms
N/A	Minimize main-thread work TBT	Main-thread busy for 777ms
N/A	Reduce the impact of third-party code TBT	Total size was 26.3MB
N/A	Eliminate render-blocking resources FCP LCP	
N/A	Avoid serving legacy JavaScript to modern browsers TBT	
N/A	Avoid large layout shifts CLS	
N/A	User Timing marks and measures	

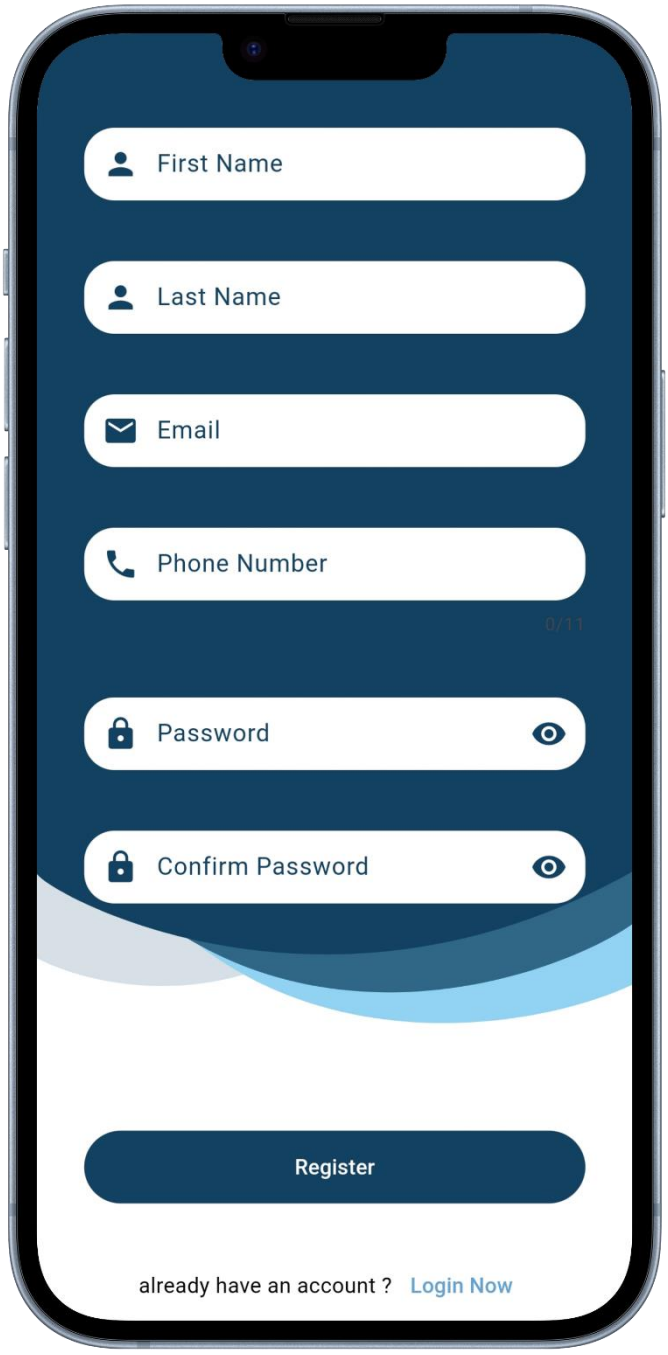
Chapter 6

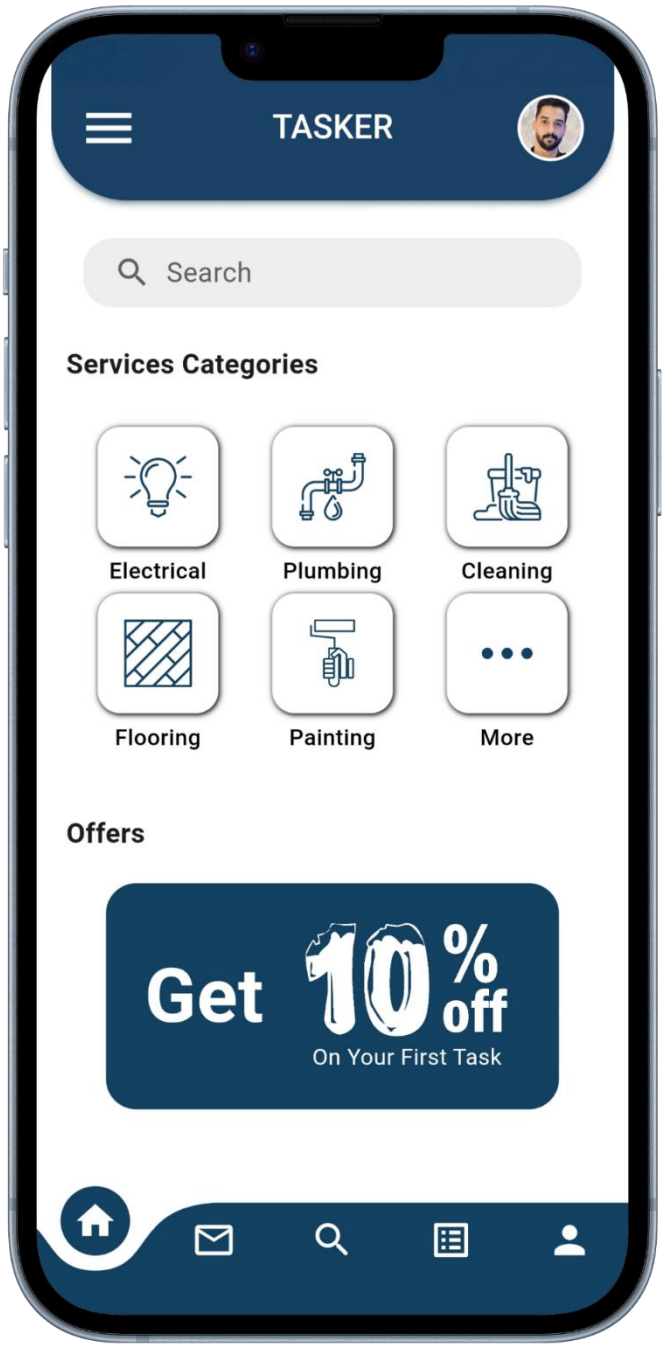
Result

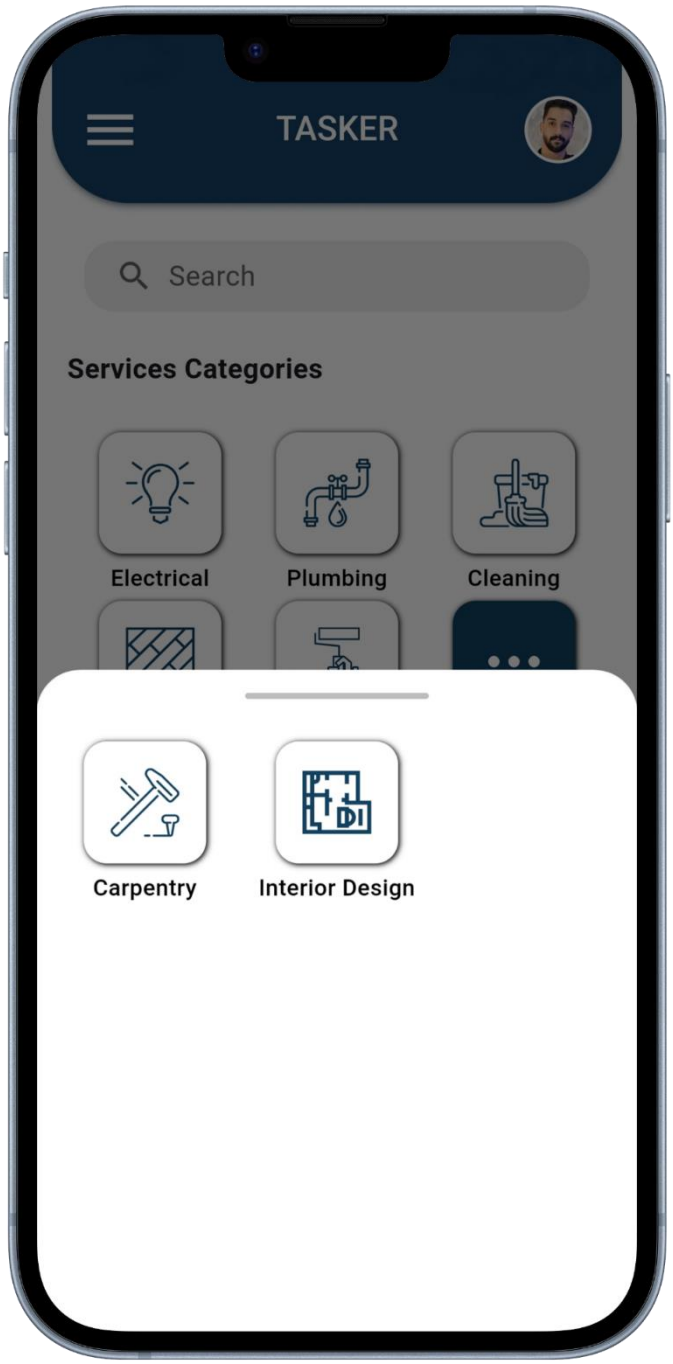
6.1 Architecture in Mobile Phone App:

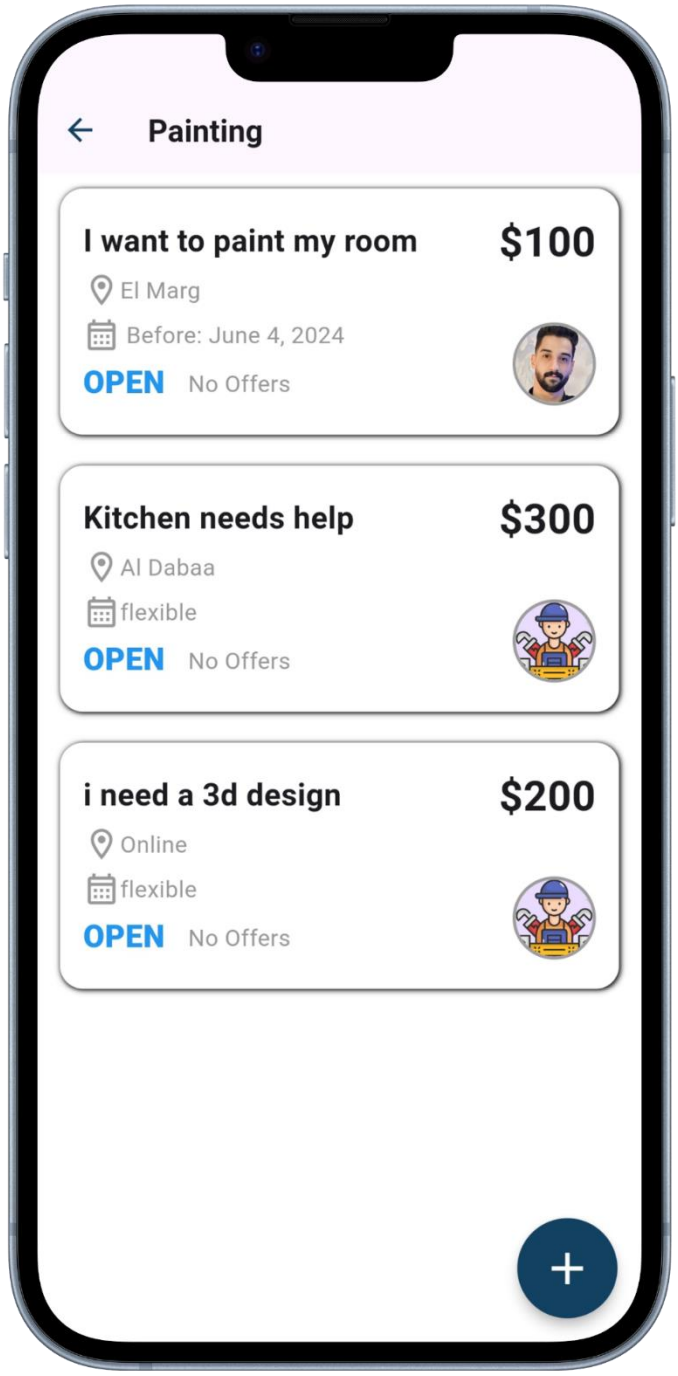

TASKER
Search
Services Categories
Cleaning
Plumbing
More
Electrical
Painting
Flooring
Offers
Get 10% off
On Your First Task

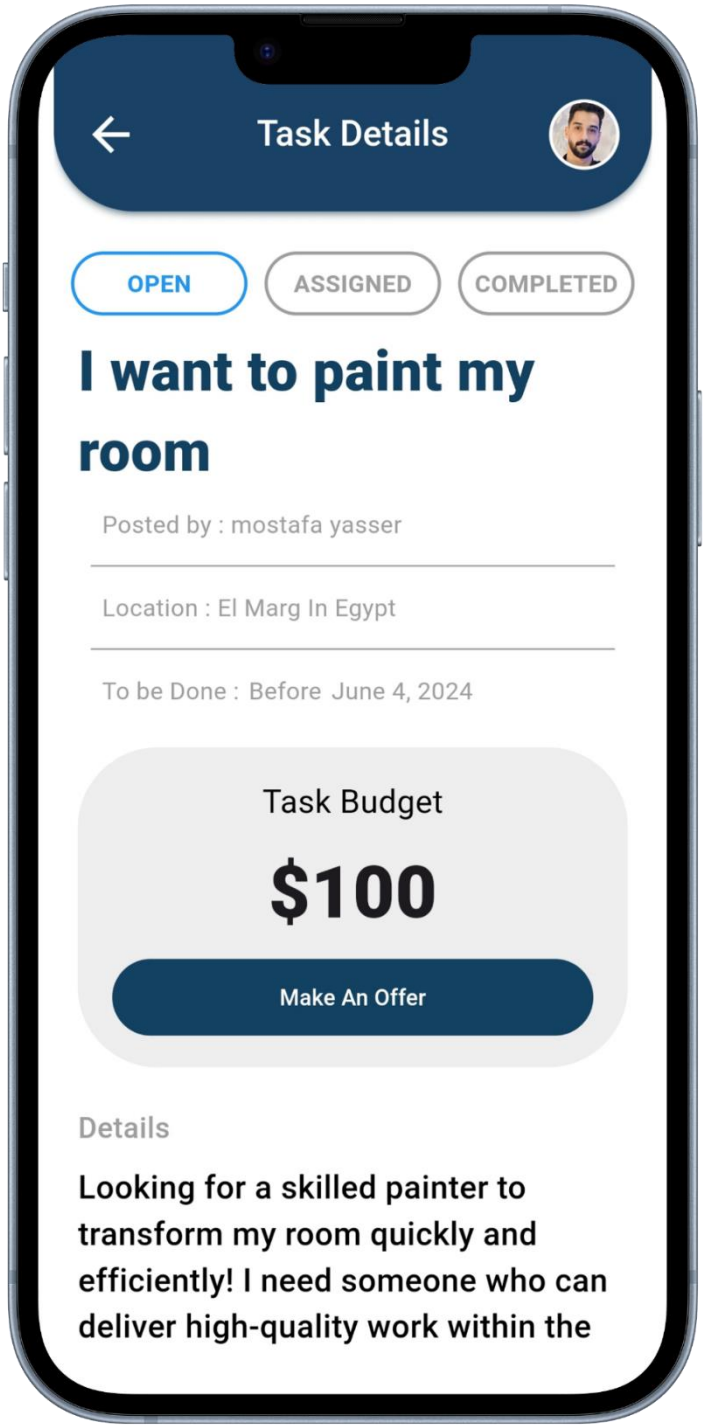

mostafa@gmail.com
Login
Don't have an account ? Register Now

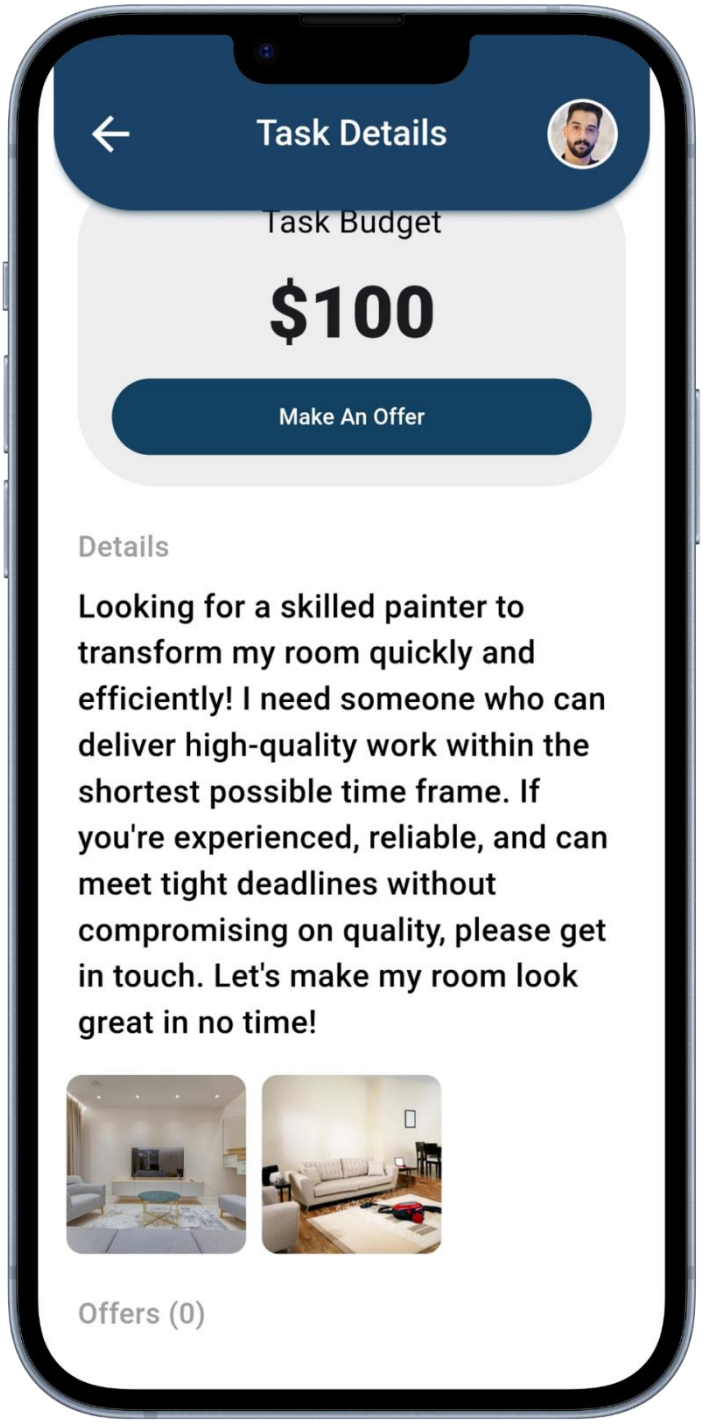

First Name
Last Name
Email
Phone Number
Password
Confirm Password
Register
already have an account ? Login Now

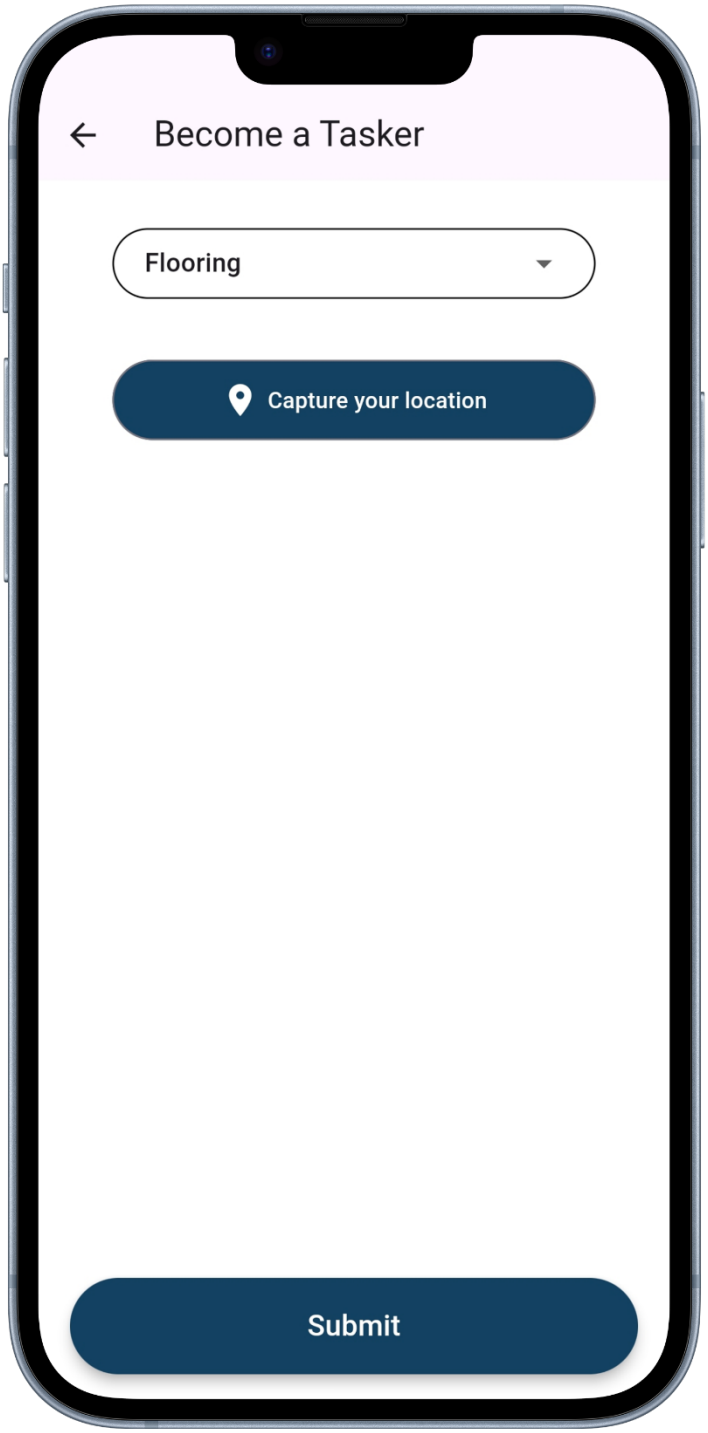

TASKER
Search
Services Categories
Electrical
Plumbing
Cleaning
Flooring
Painting
More
Offers
Get 10% off
On Your First Task


TASKER
Search
Services Categories
Electrical
Plumbing
Cleaning
Carpentry
Interior Design


Painting
I want to paint my room
$100
El Marg
Before: June 4, 2024
OPEN
No Offers
Kitchen needs help
$300
Al Dabaa
flexible
OPEN
No Offers
i need a 3d design
$200
Online
flexible
OPEN
No Offers

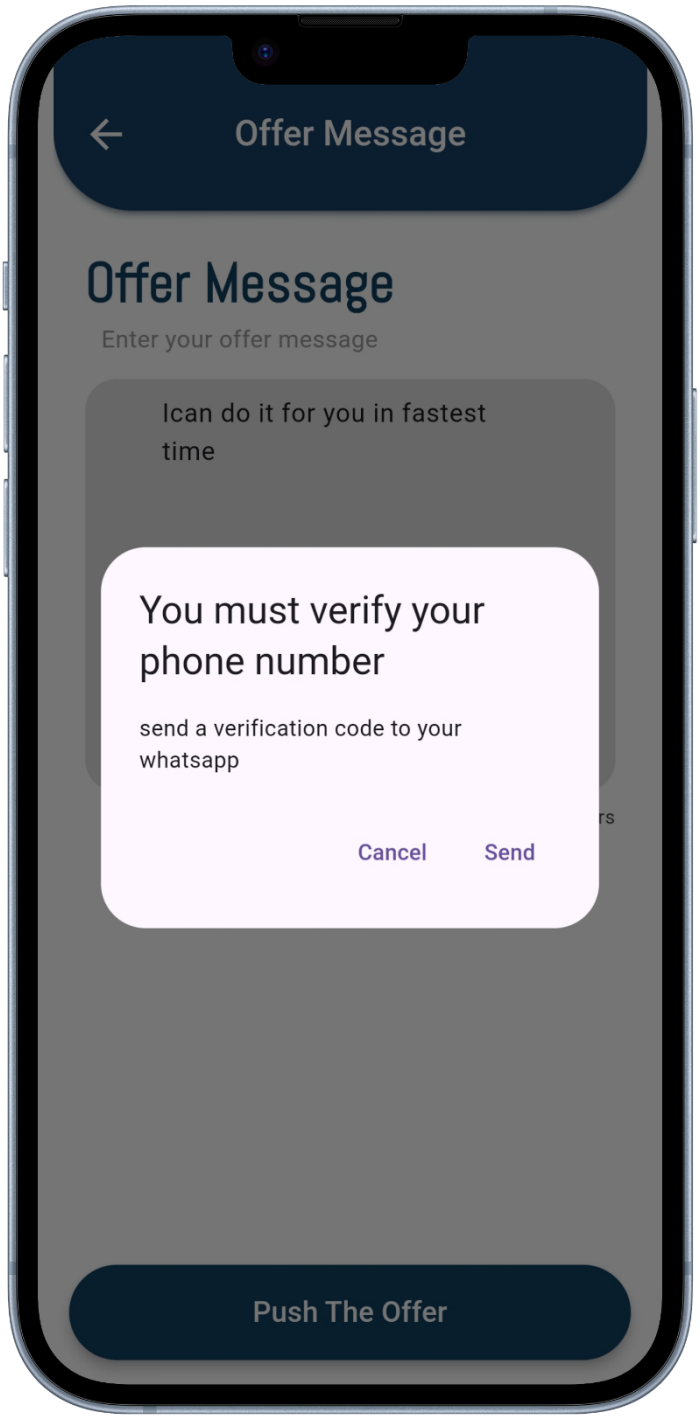

Task Details
OPEN
ASSIGNED
COMPLETED
I want to paint my room
Posted by : mostafa yasser
Location : El Marg In Egypt
To be Done : Before June 4, 2024
Task Budget
$100
Make An Offer
Details
Looking for a skilled painter to transform my room quickly and efficiently! I need someone who can deliver high-quality work within the

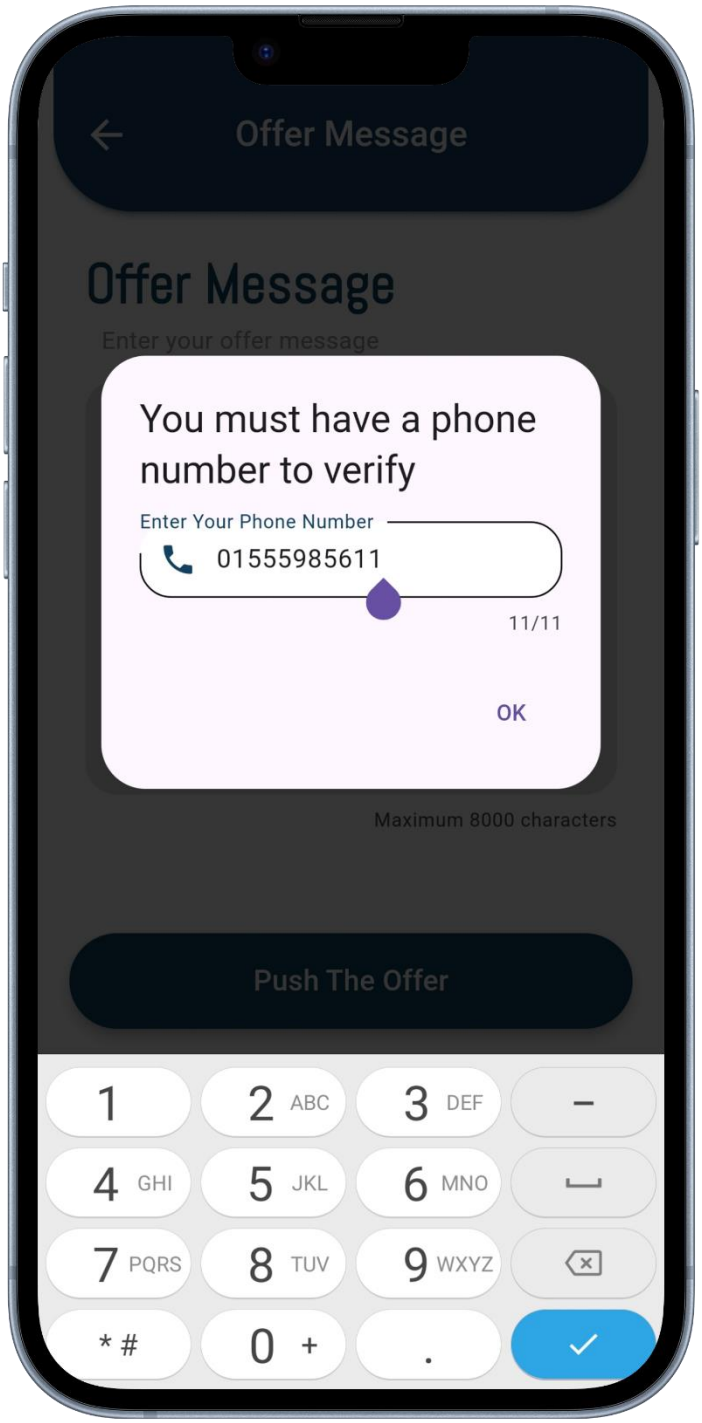

Task Details
Task Budget
$100
Make An Offer
Details
Looking for a skilled painter to transform my room quickly and efficiently! I need someone who can deliver high-quality work within the shortest possible time frame. If you're experienced, reliable, and can meet tight deadlines without compromising on quality, please get in touch. Let's make my room look great in no time!
Offers (0)


Become a Tasker
Flooring
Capture your location
Submit

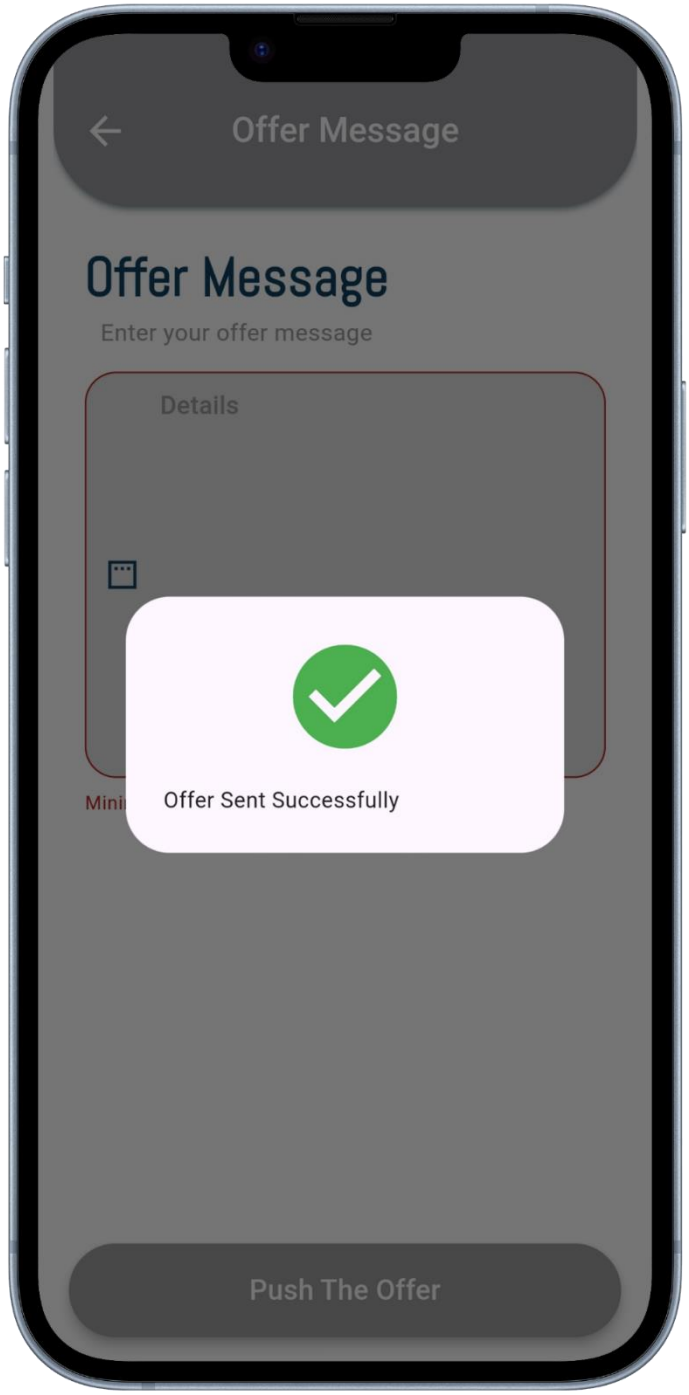

Make An Offer
Enter Your Offer
Enter your offer amount
$80
1
2
3
4
5
6
7
8
9
0
Continue

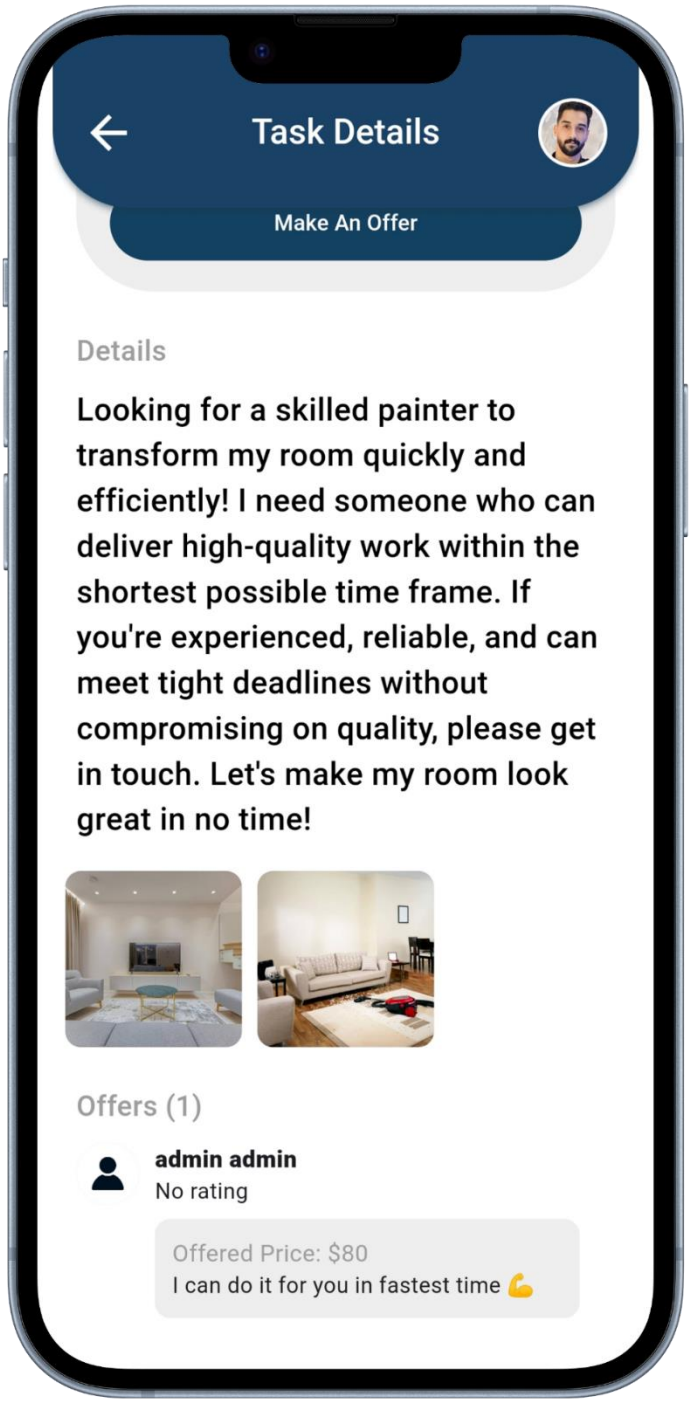

Offer Message
Offer Message
Enter your offer message
I can do it for you in fastest time 💪
Maximum 8000 characters
Push The Offer

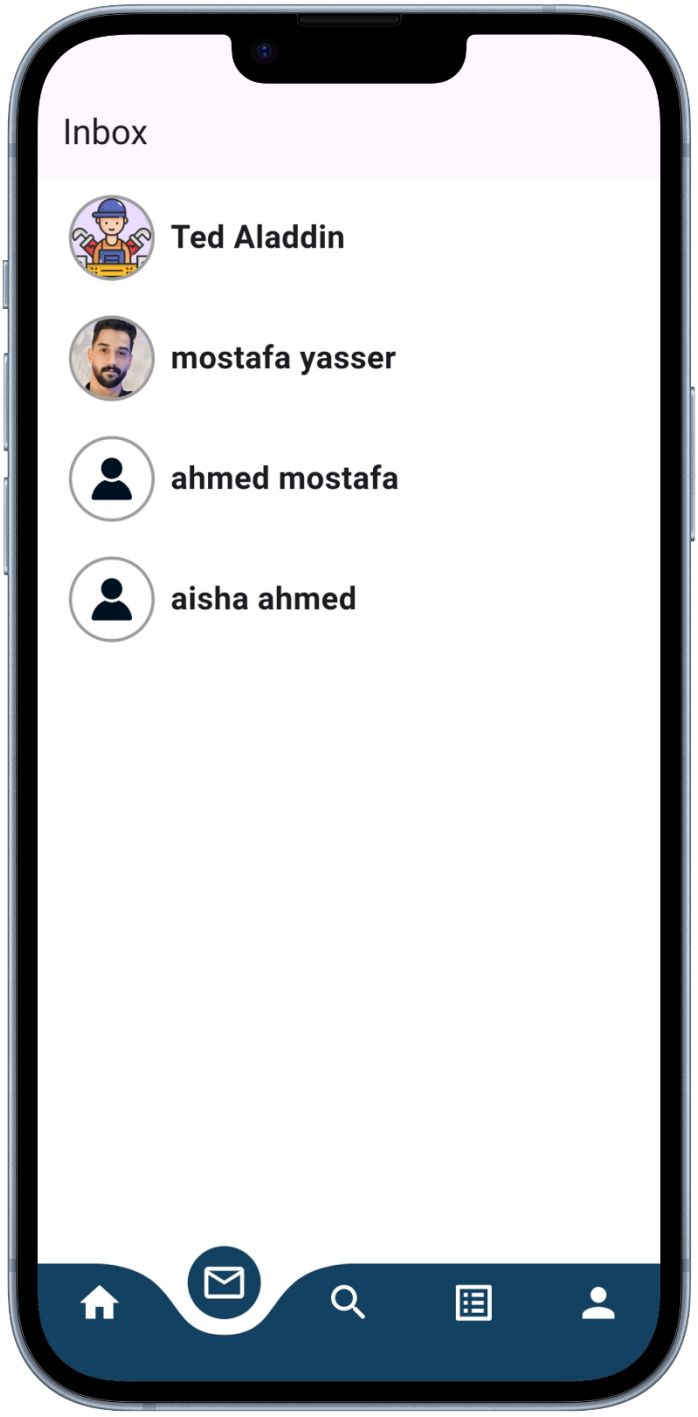

Offer Message
Offer Message
Enter your offer message
Ican do it for you in fastest time
You must verify your phone number
send a verification code to your whatsapp
Cancel
Send
Push The Offer

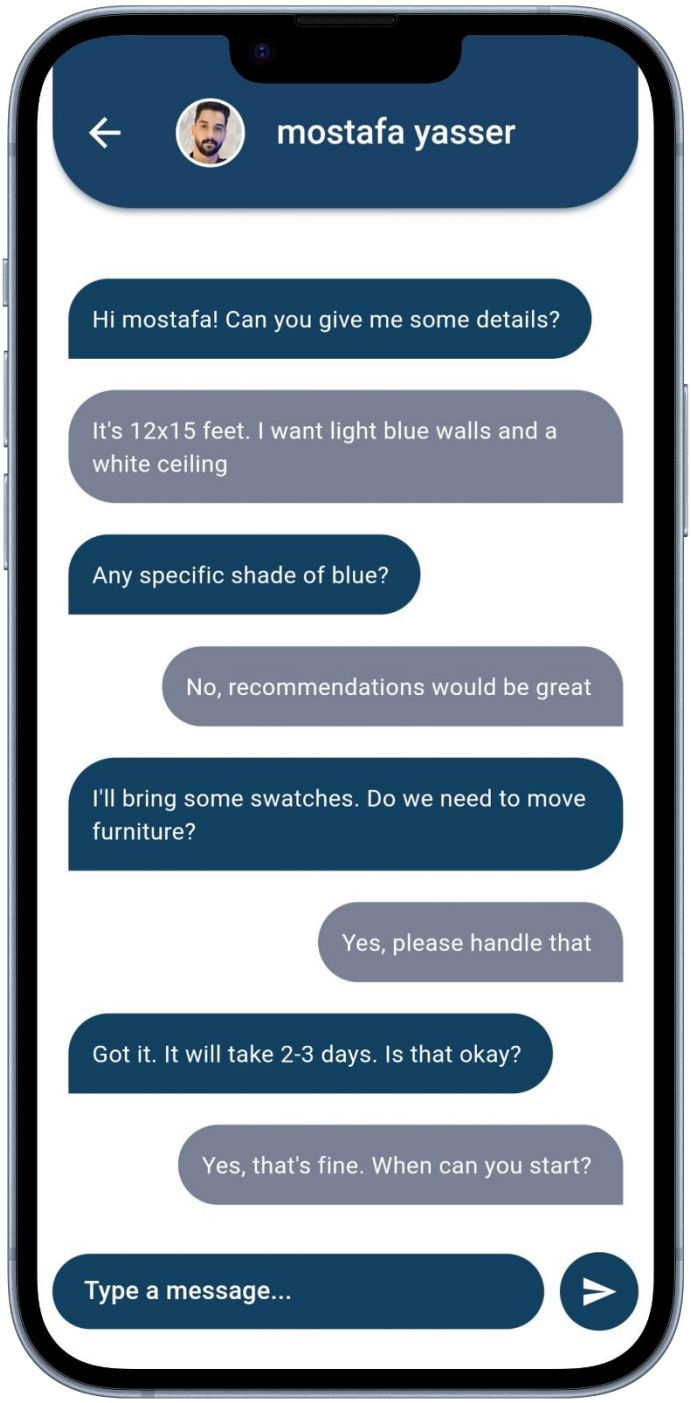

Offer Message
Offer Message
Enter your offer message
You must have a phone number to verify
Enter Your Phone Number
01555985611
11/11
OK
Maximum 8000 characters
Push The Offer

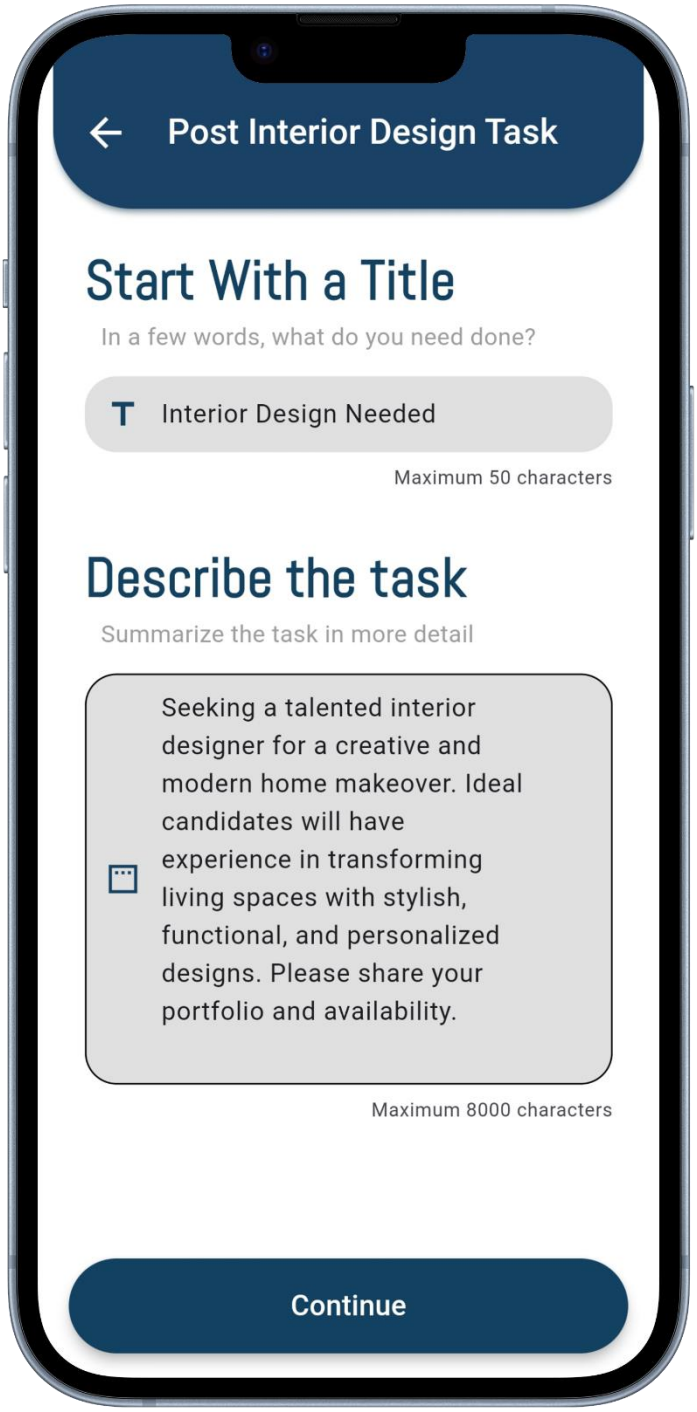

Phone Number Verification
Verification Code
Enter the 6-digit code sent to your WhatsApp number
1
2
8
4
4
9
Resend in 21s
Confirm

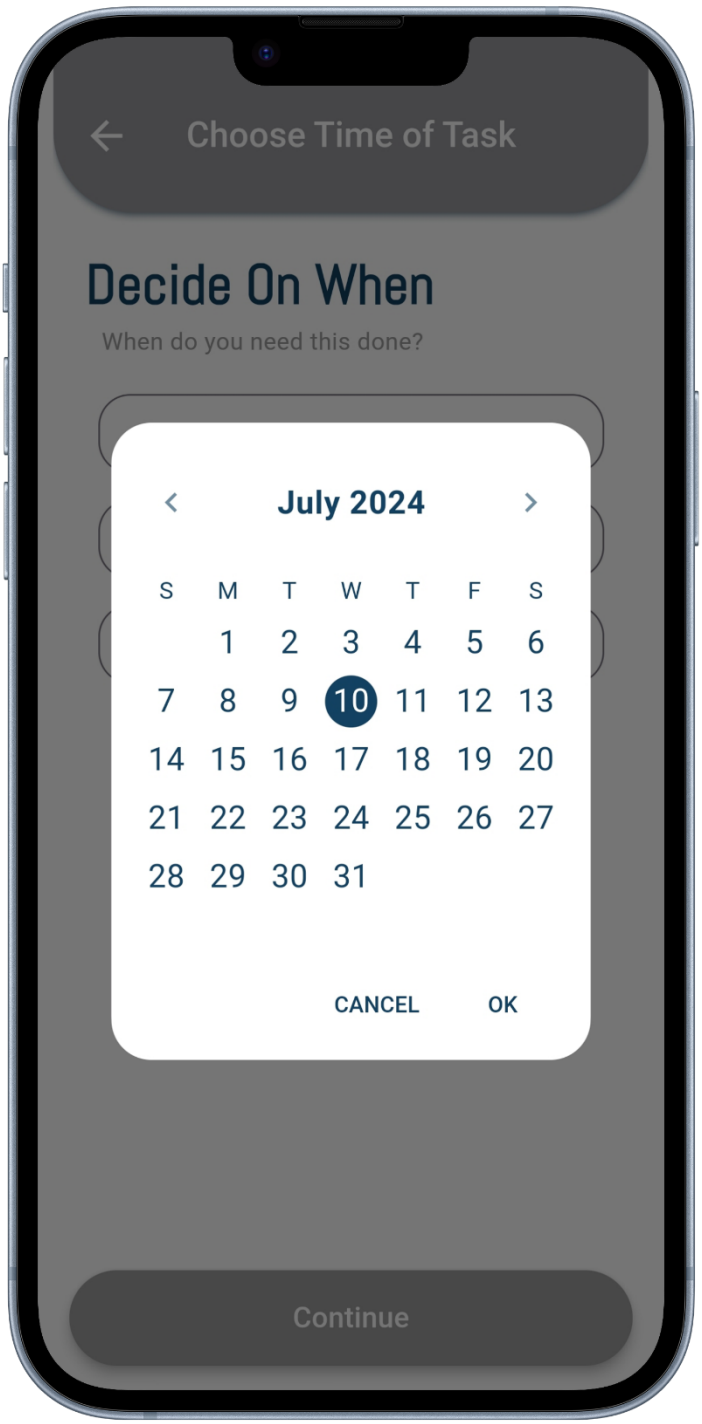

Offer Message
Offer Message
Enter your offer message
Details
Offer Sent Successfully
Push The Offer

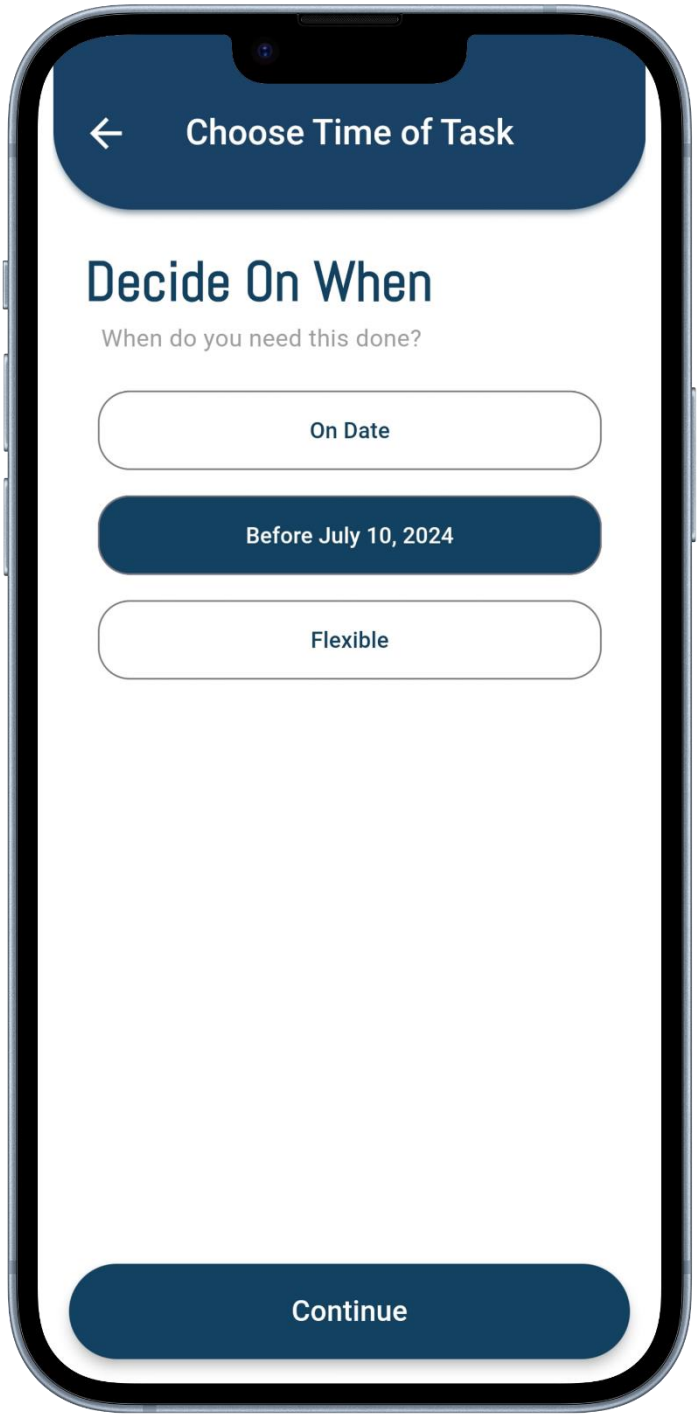

Task Details
Make An Offer
Details
Looking for a skilled painter to transform my room quickly and efficiently! I need someone who can deliver high-quality work within the shortest possible time frame. If you're experienced, reliable, and can meet tight deadlines without compromising on quality, please get in touch. Let's make my room look great in no time!
Offers (1)
admin admin
No rating
Offered Price: $80
I can do it for you in fastest time 💪

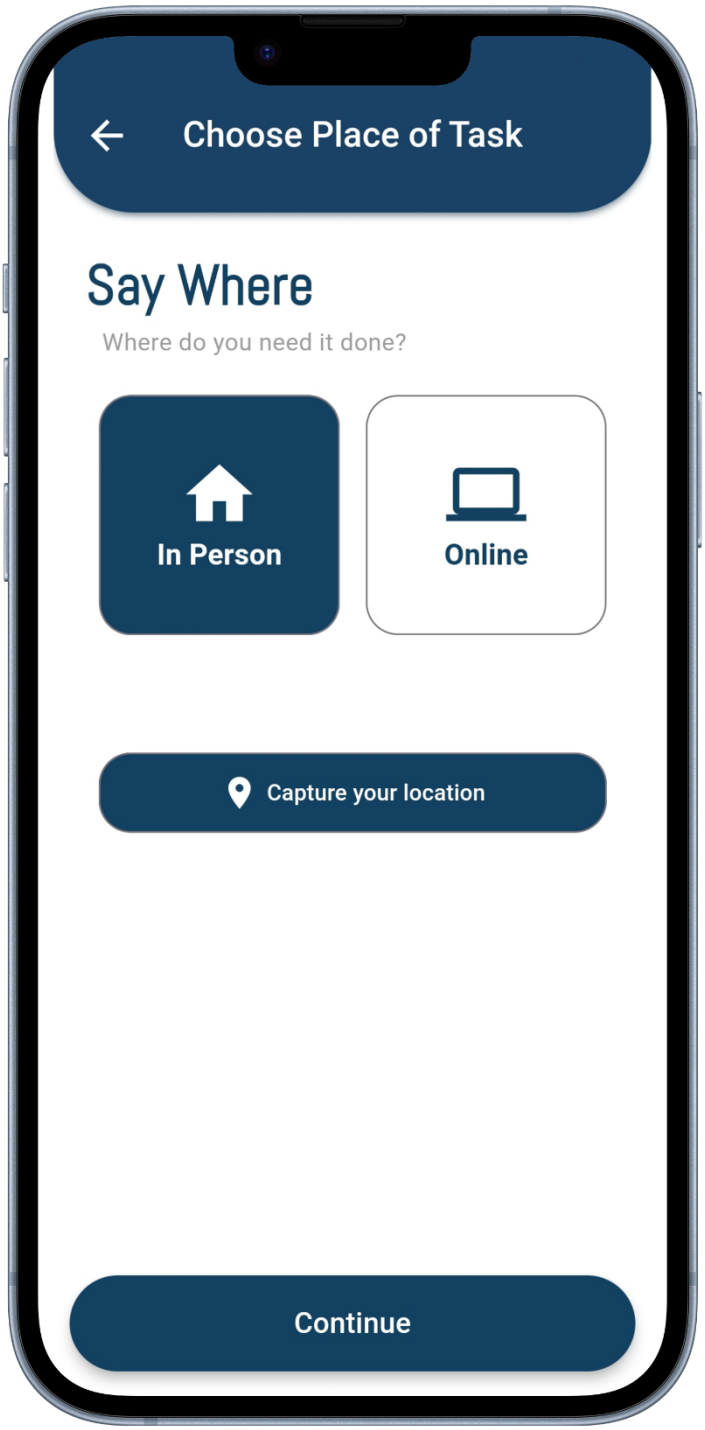

Inbox
Ted Aladdin
mostafa yasser
ahmed mostafa
aisha ahmed

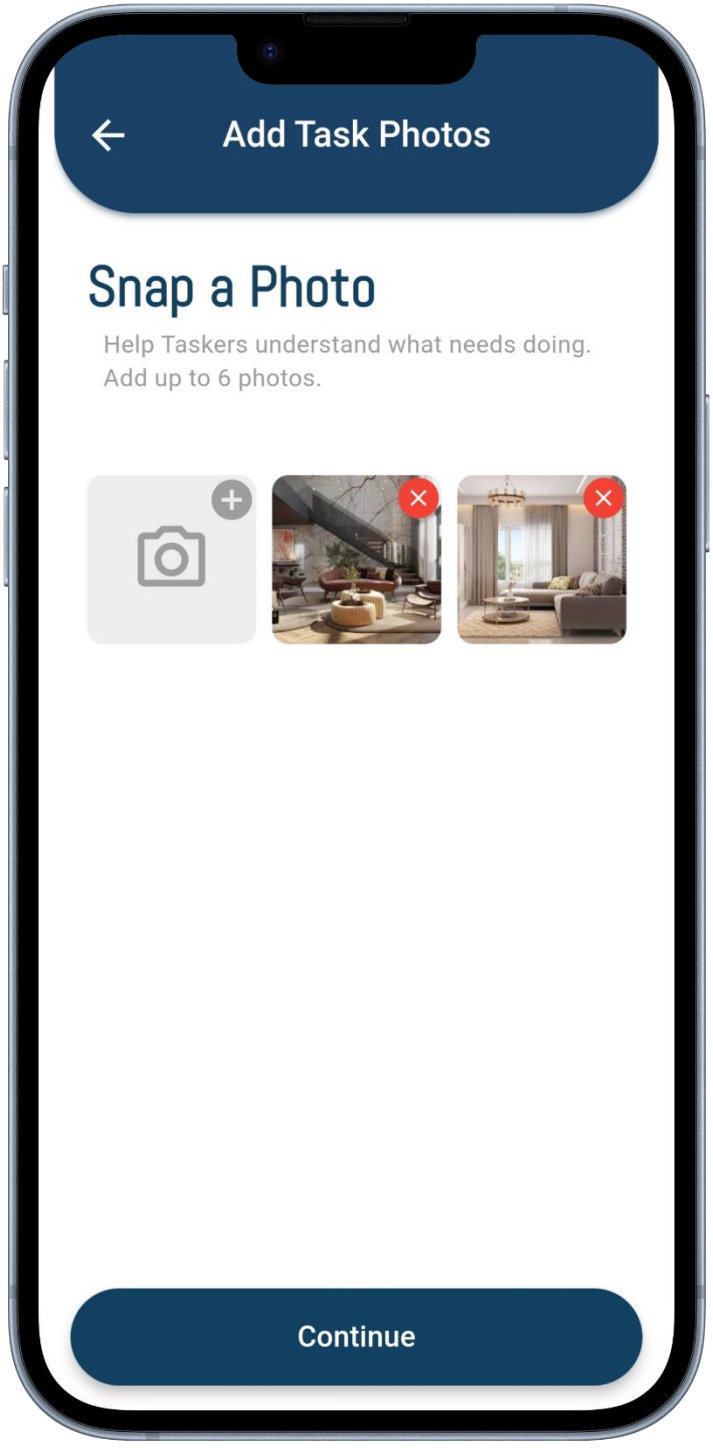

mostafa yasser
Hi mostafa! Can you give me some details?
It's 12x15 feet. I want light blue walls and a white ceiling
Any specific shade of blue?
No, recommendations would be great
I'll bring some swatches. Do we need to move furniture?
Yes, please handle that
Got it. It will take 2-3 days. Is that okay?
Yes, that's fine. When can you start?
Type a message...

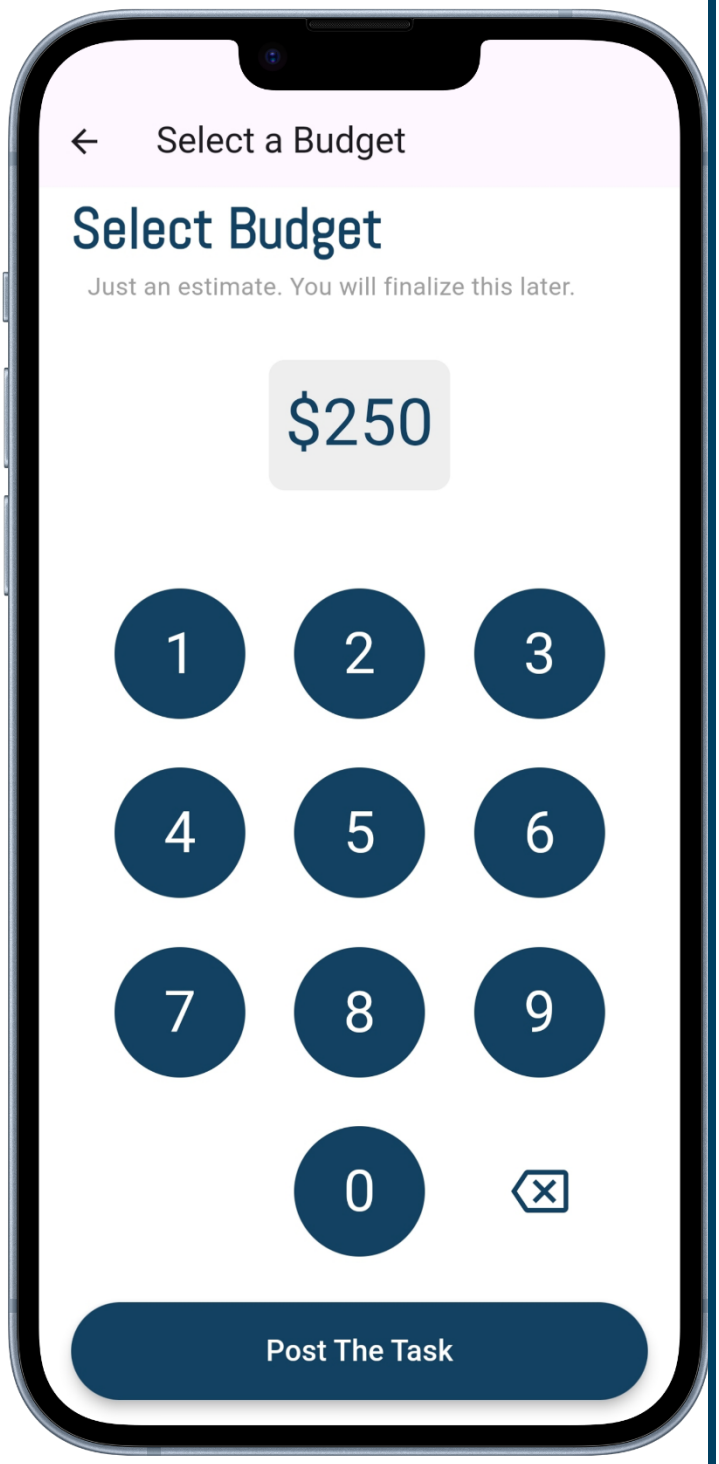

Post Interior Design Task
Start With a Title
In a few words, what do you need done?
T
Interior Design Needed
Maximum 50 characters
Describe the task
Summarize the task in more detail
Seeking a talented interior designer for a creative and modern home makeover. Ideal candidates will have experience in transforming living spaces with stylish, functional, and personalized designs. Please share your portfolio and availability.
Maximum 8000 characters
Continue

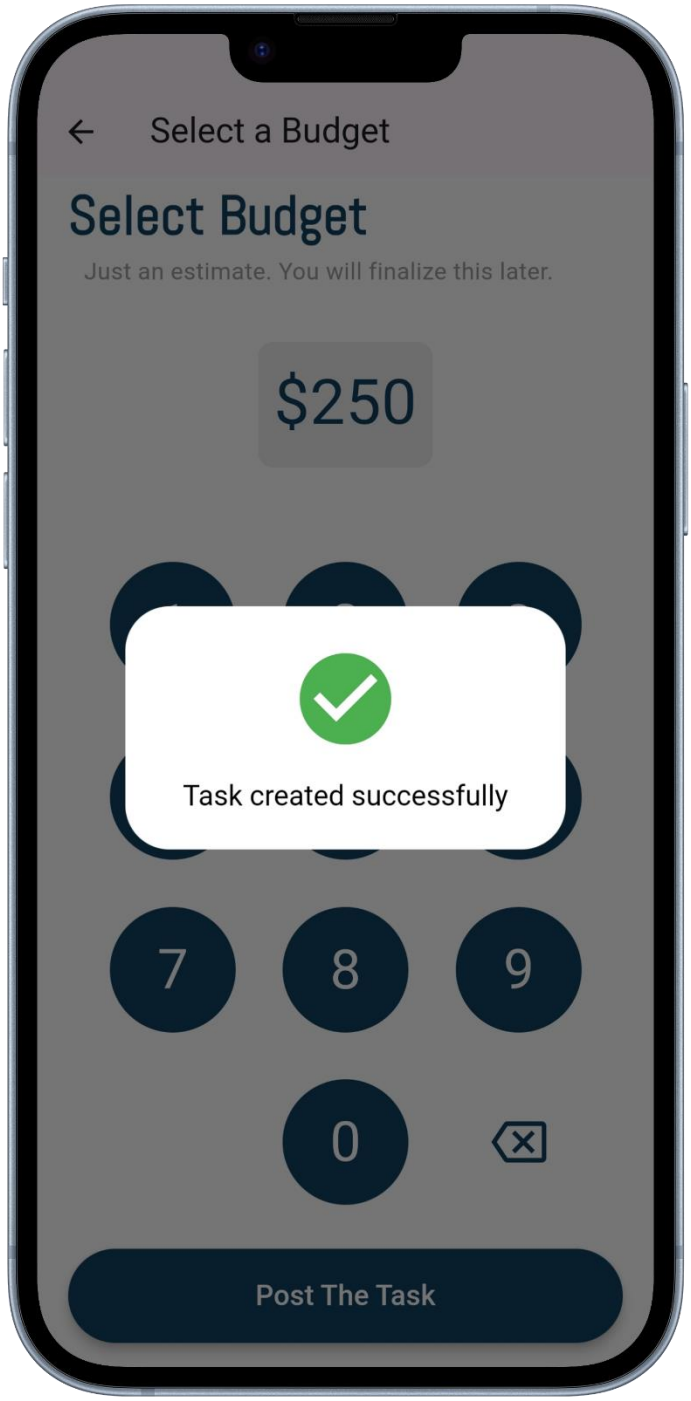

Choose Time of Task
Decide On When
When do you need this done?
July 2024
S M T W T F S
1 2 3 4 5 6
7 8 9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30 31
CANCEL
OK
Continue


Choose Time of Task
Decide On When
When do you need this done?
On Date
Before July 10, 2024
Flexible
Continue

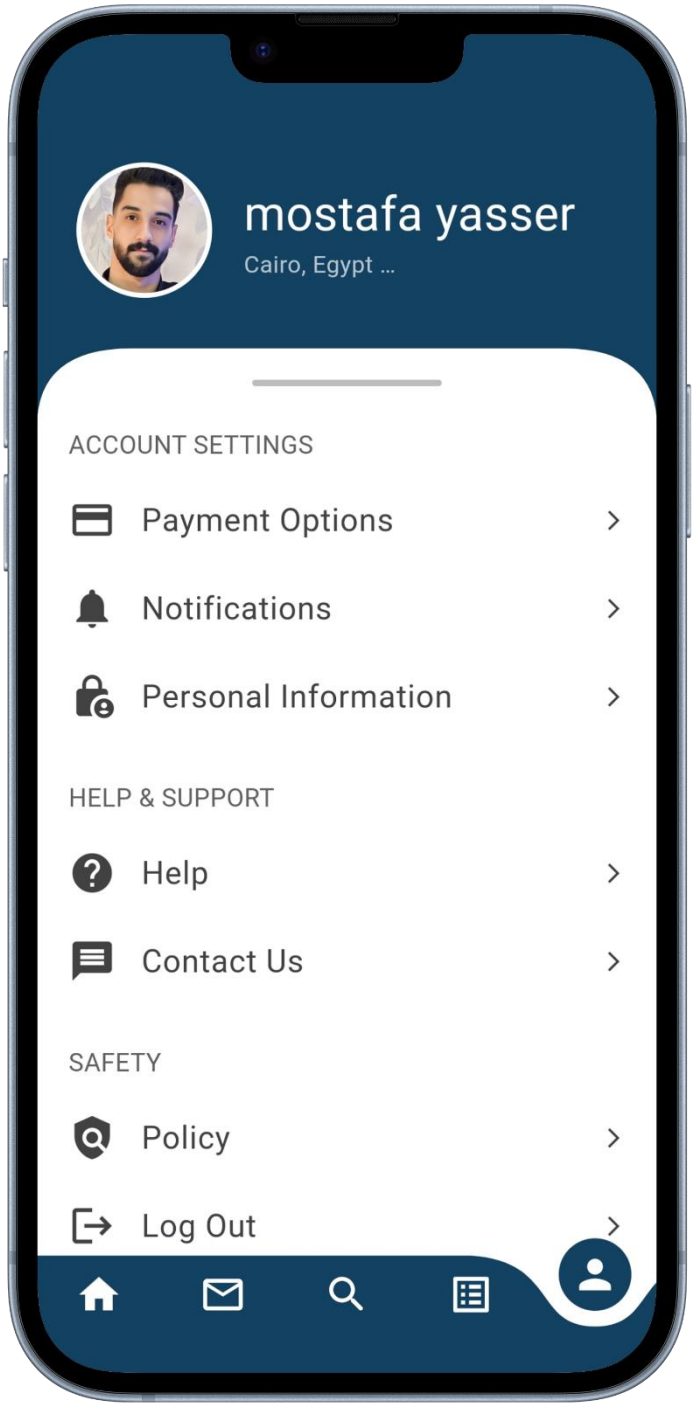

Choose Place of Task
Say Where
Where do you need it done?
In Person
Online
Capture your location
Continue

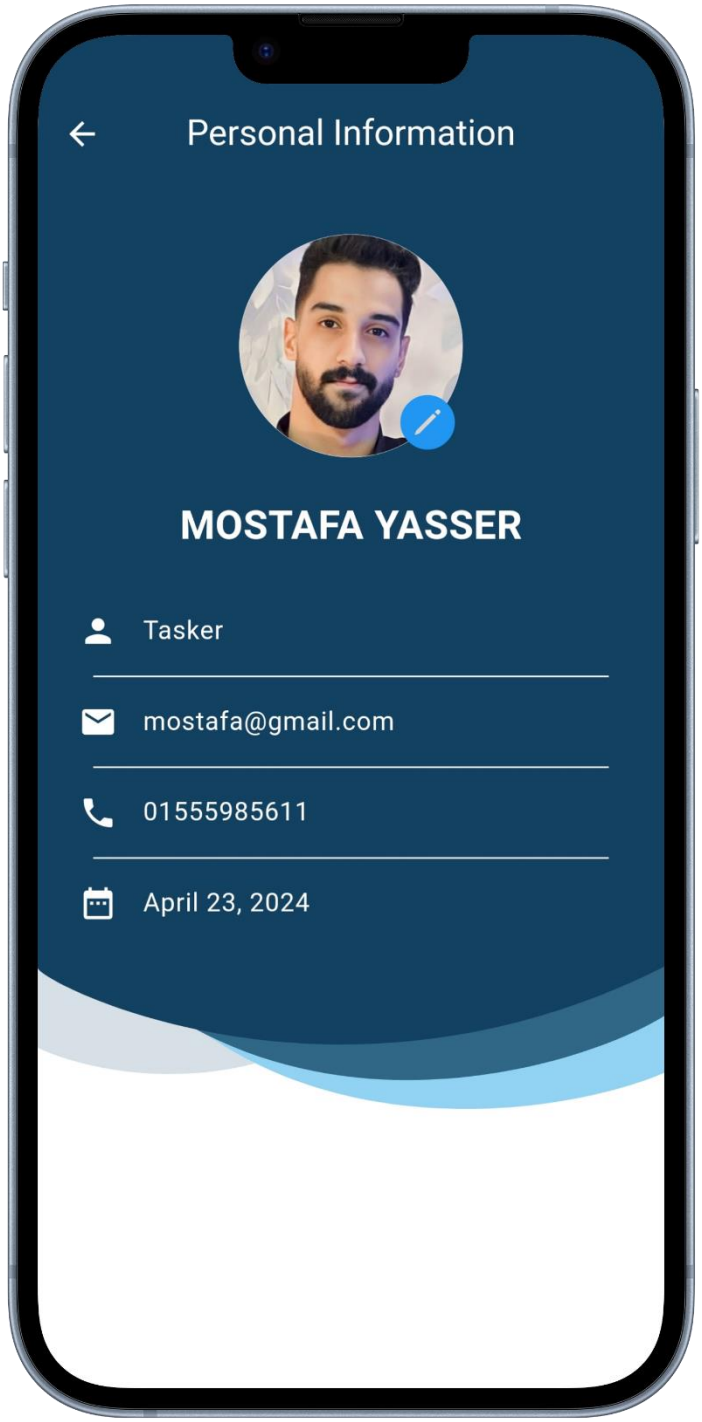

Add Task Photos
Snap a Photo
Help Taskers understand what needs doing.
Add up to 6 photos.
Continue

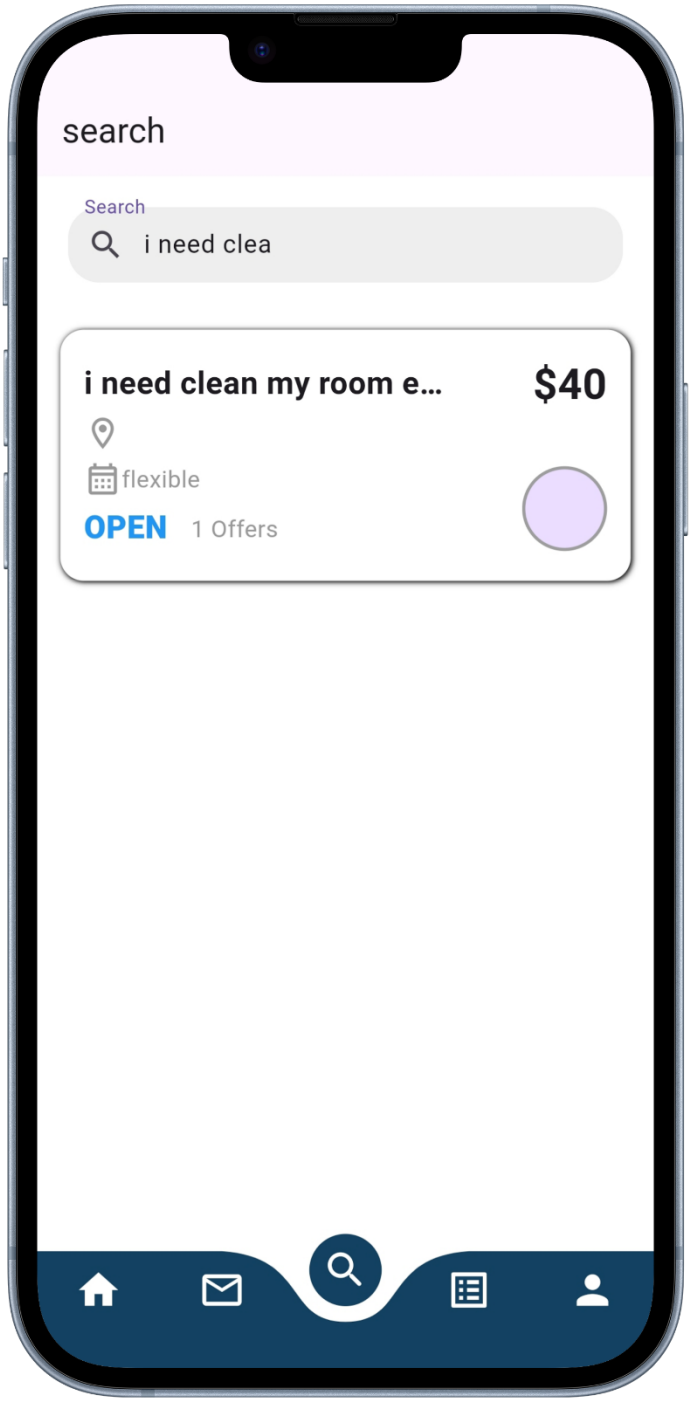

Select a Budget
Select Budget
Just an estimate. You will finalize this later.
$250
1
2
3
4
5
6
7
8
9
0
Post The Task

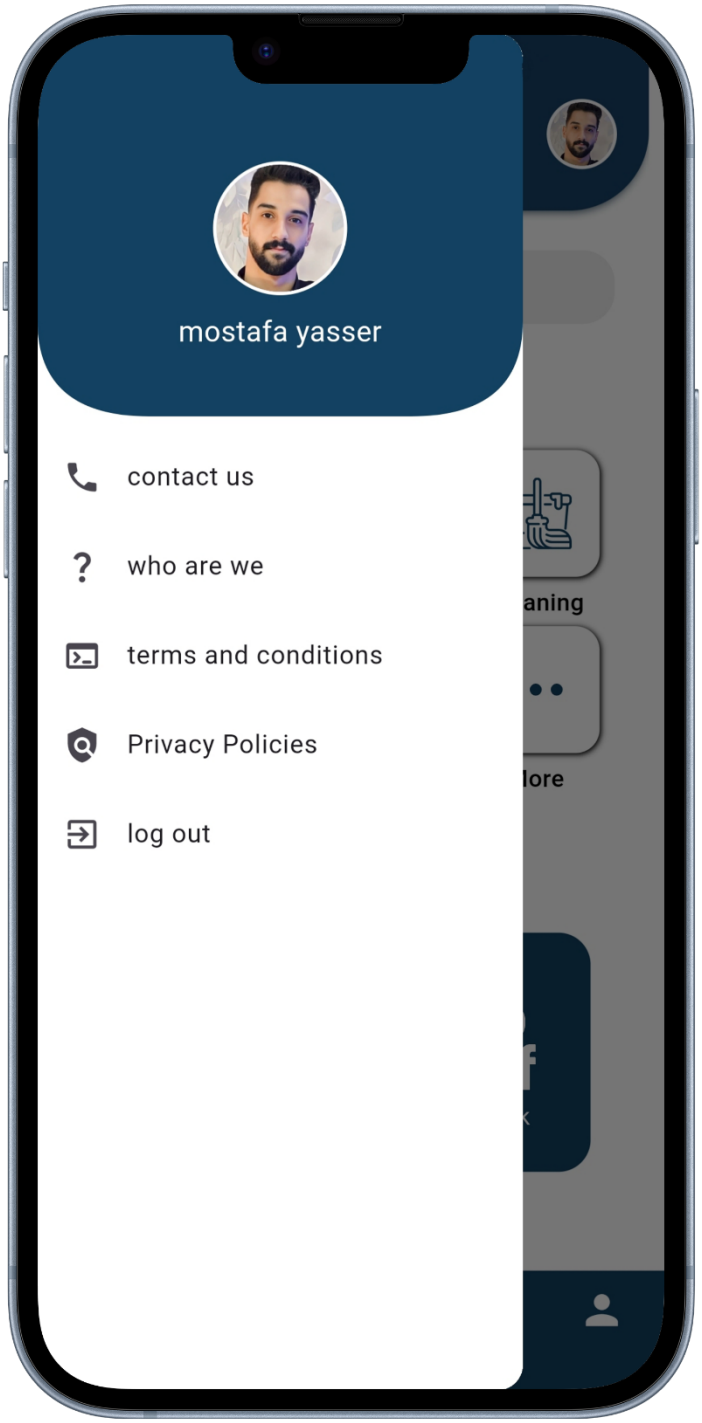

Select a Budget
Select Budget
Just an estimate. You will finalize this later.
$250
Task created successfully
7
8
9
0
Post The Task


8:35 Thursday, June 06
Fix Flex • 14m
New Task
A new task is available

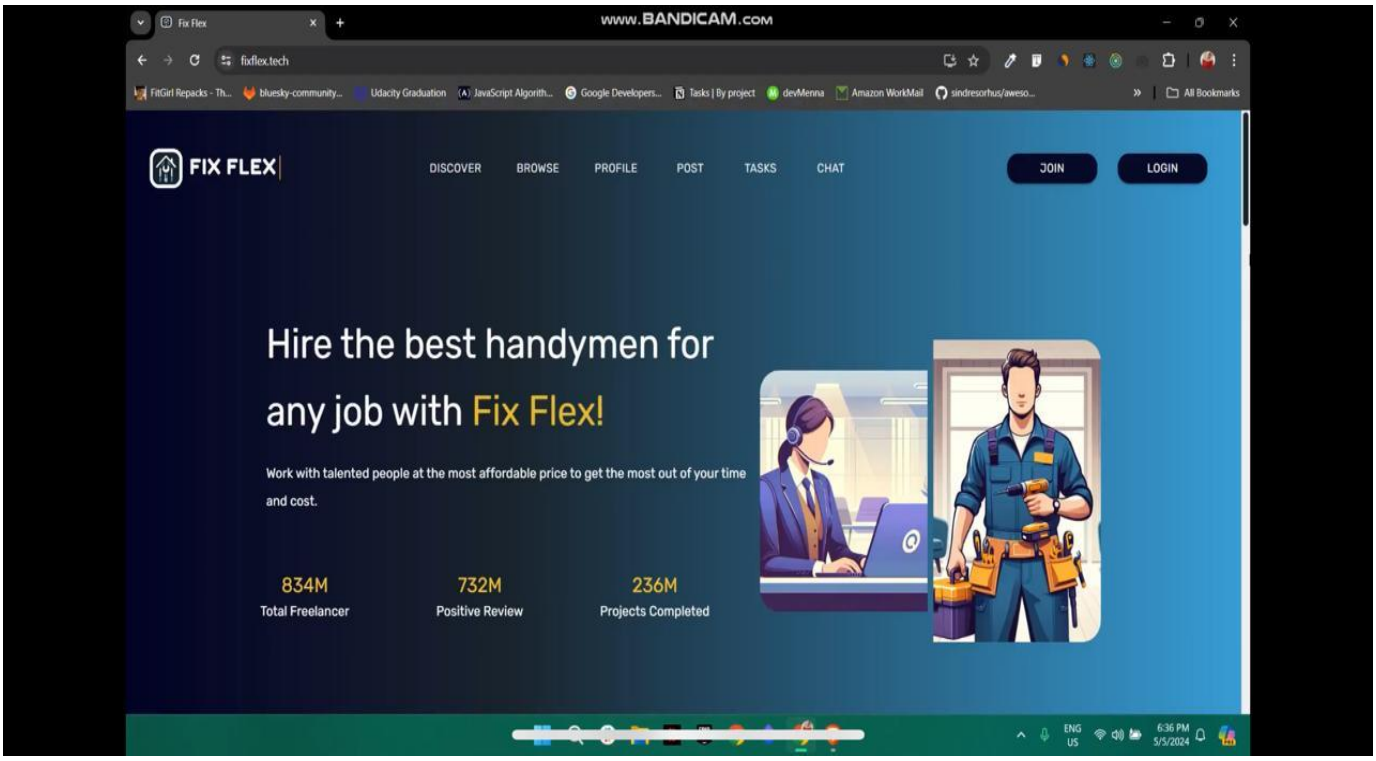

mostafa yasser
Cairo, Egypt ...
ACCOUNT SETTINGS
Payment Options
Notifications
Personal Information
HELP & SUPPORT
Help
Contact Us
SAFETY
Policy
Log Out

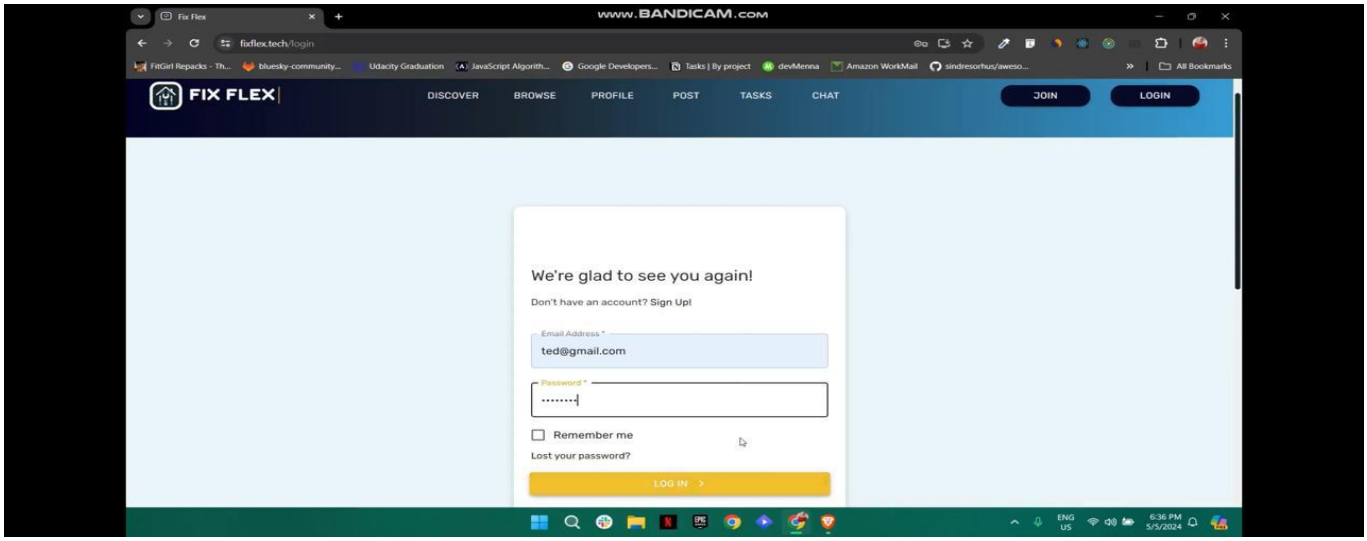

Personal Information
MOSTAFA YASSER
Tasker
mostafa@gmail.com
01555985611
April 23, 2024

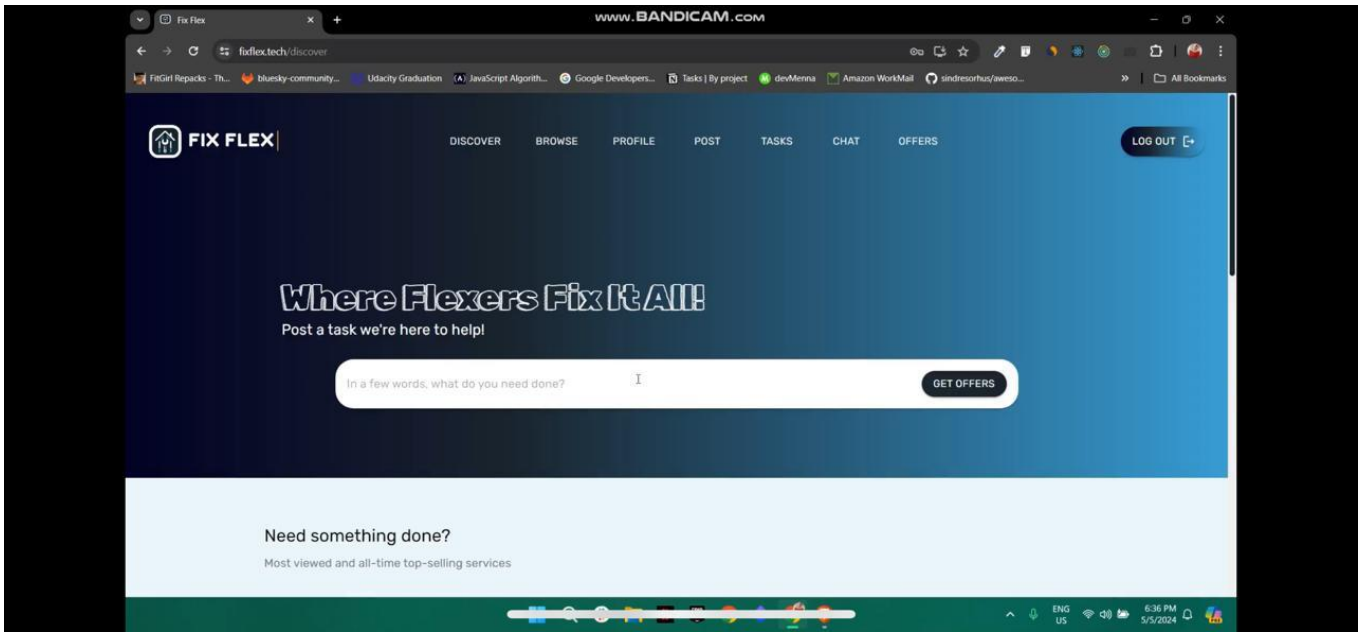

search
Search
i need clea
i need clean my room e...
$40
flexible
OPEN
1 Offers

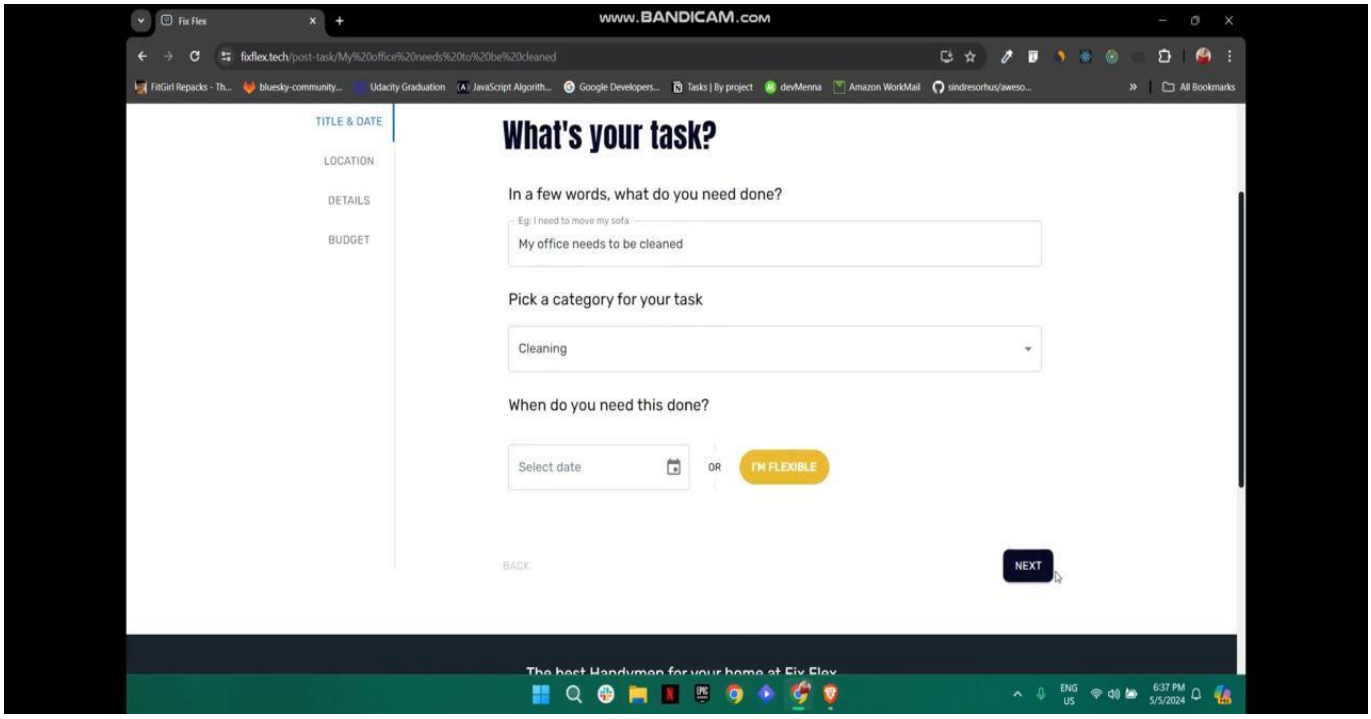

mostafa yasser
contact us
who are we
terms and conditions
Privacy Policies
log out

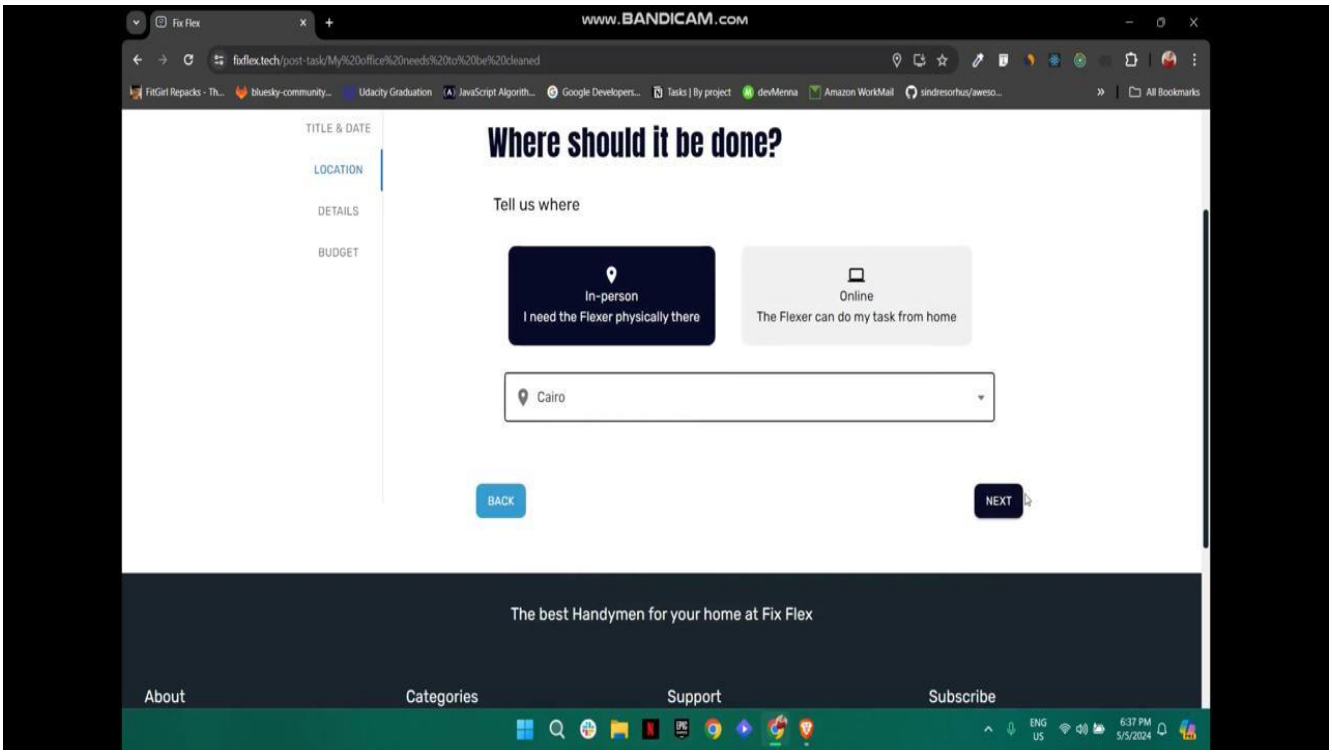
Architecture in Website:

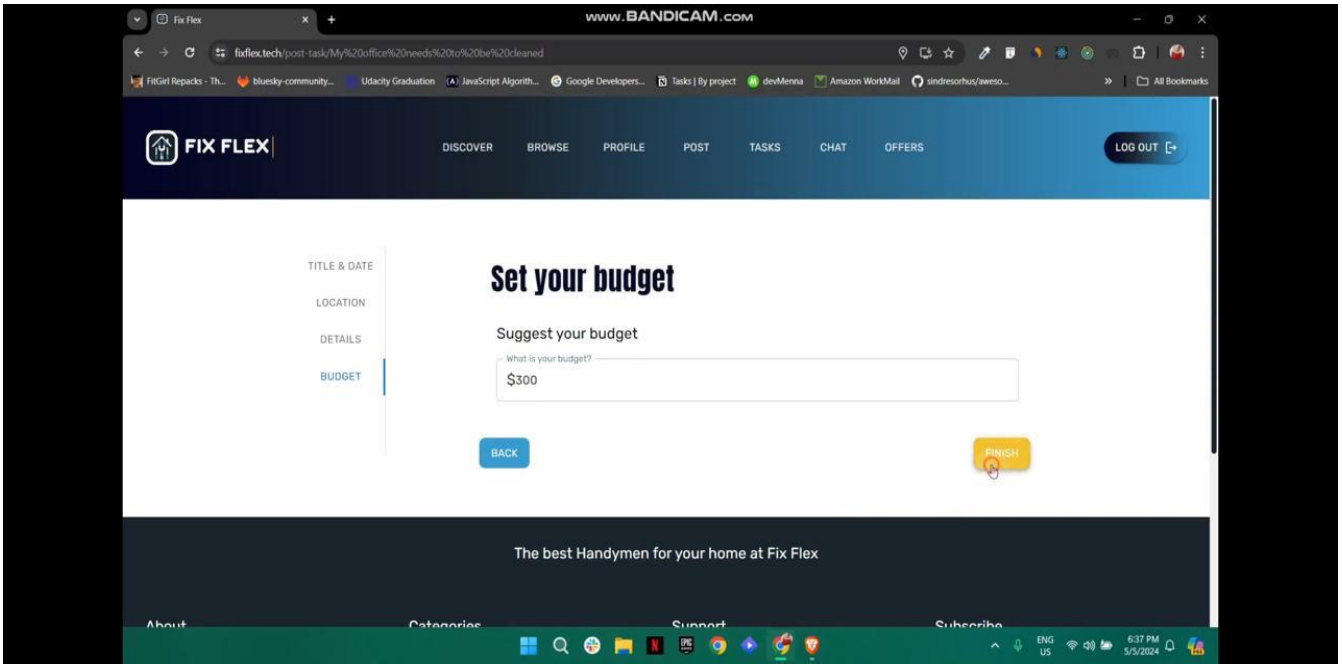


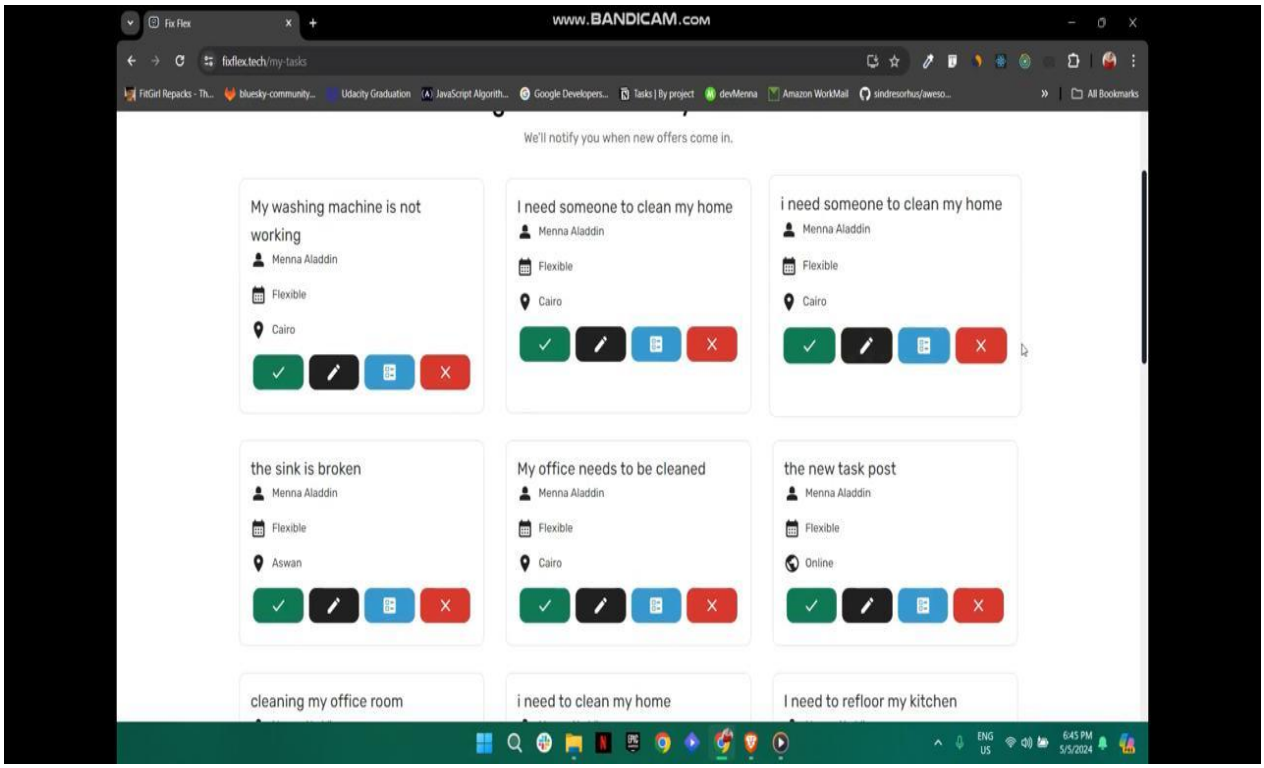



www.BANDICAM.com
fixflex.tech/discover
FIX FLEX
DISCOVER
BROWSE
PROFILE
POST
TASKS
CHAT
OFFERS
LOG OUT
Where Flexers Fix It All!
Post a task we're here to help!
In a few words, what do you need done?
GET OFFERS
Need something done?
Most viewed and all-time top-selling services


www.BANDICAM.com
fixflex.tech/post-task/My%20office%20needs%20to%20be%20cleaned
TITLE & DATE
LOCATION
DETAILS
BUDGET
What's your task?
In a few words, what do you need done?
Eg: I need to move my sofa
My office needs to be cleaned
Pick a category for your task
Cleaning
When do you need this done?
Select date
OR
I'M FLEXIBLE
BACK
NEXT


www.BANDICAM.com
fixflex.tech/post-task/My%20office%20needs%20to%20be%20cleaned
TITLE & DATE
LOCATION
DETAILS
BUDGET
Where should it be done?
Tell us where
In-person
I need the Flexer physically there
Online
The Flexer can do my task from home
Cairo
BACK
NEXT
The best Handymen for your home at Fix Flex
About
Categories
Support
Subscribe


www.BANDICAM.com
fixflex.tech/post-task/My%20office%20needs%20to%20be%20cleaned
FIX FLEX
DISCOVER
BROWSE
PROFILE
POST
TASKS
CHAT
OFFERS
LOG OUT
TITLE & DATE
LOCATION
DETAILS
BUDGET
Set your budget
Suggest your budget
What is your budget?
$300
BACK
FINISH
The best Handymen for your home at Fix Flex


www.BANDICAM.com
Fix Flex
fixflex.tech/my-tasks
We'll notify you when new offers come in.
My washing machine is not working
Menna Aladdin
Flexible
Cairo
I need someone to clean my home
Menna Aladdin
Flexible
Cairo
i need someone to clean my home
Menna Aladdin
Flexible
Cairo
the sink is broken
Menna Aladdin
Flexible
Aswan
My office needs to be cleaned
Menna Aladdin
Flexible
Cairo
the new task post
Menna Aladdin
Flexible
Online
cleaning my office room
i need to clean my home
I need to refloor my kitchen

Chapter 7

future steps and Conclusion

7.1 Final reflections and future steps:

• At FIX FLEX, we are committed to continuously improving our platform to meet the evolving needs of

our users.

• By listening to our community and implementing their feedback, we aim to provide a seamless and reliable experience for both users and taskers.

• With a roadmap full of exciting features and enhancements, we are dedicated to making FIX FLEX the go-to platform for all household task needs.

7.2 Conclusion

In conclusion, the FixFlex platform represents a significant advancement in connecting individuals seeking various task services with skilled professionals ready to offer their expertise. By prioritizing accessibility, reliability, and community support, FixFlex aims to become a go-to solution for everyday tasks and services.

We extend our heartfelt gratitude to Dr. Helal Ahmed, our esteemed supervisor, for his invaluable guidance and support throughout our graduation project. His insights and encouragement have been instrumental in the successful development and implementation of FixFlex.

Download mobile application: FixFlexDemo8.apk

FIX FLEX web application:
https://www.fixflex.tech/

Email: info@fixflex.tech

GitHub: https://github.com/fixflex