

DESIGN DOCUMENTATION

Easify App

Submitted By: Akshara R

Sufirath V S

CONTENTS

1.Usestory	
2.Architecture	
3.Technology 3.1 front end 3.2 Back end 3.3 database	
4.Wireframes	
5.API	
6.JIRA	
7.GIT	
8.MILESTONE	

1.USER STORY

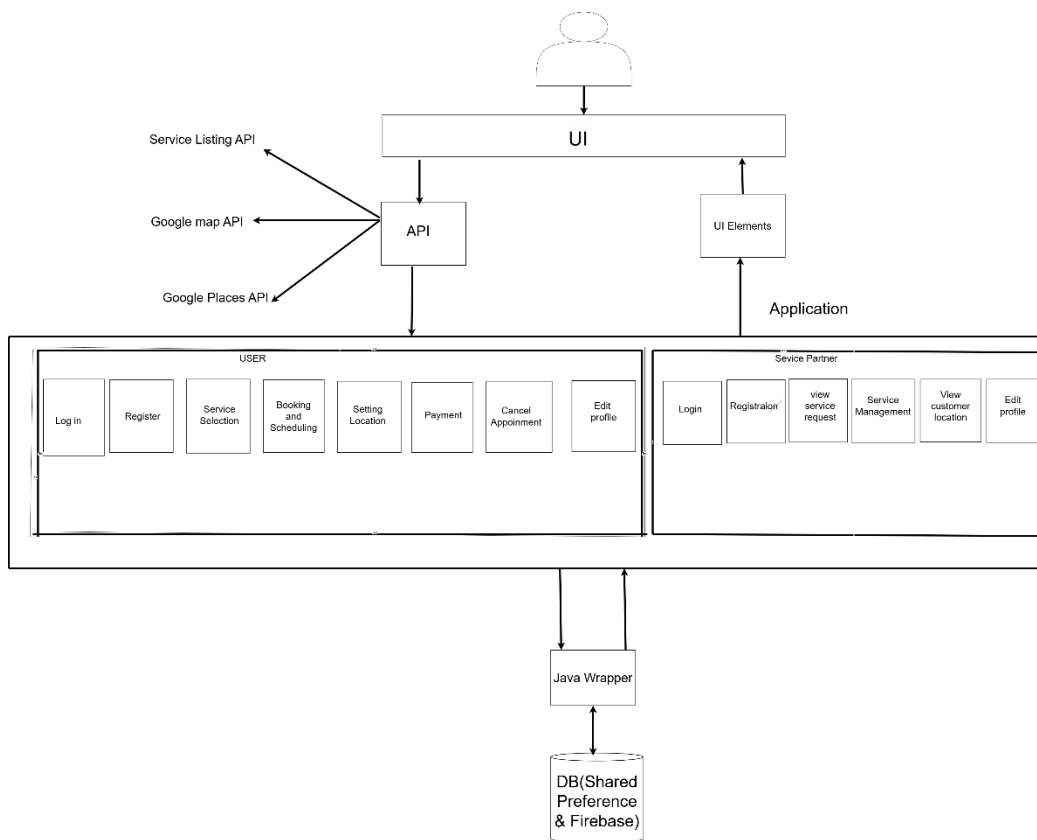
User will have the following functionality

- Registration and Login
- Service Selection
- Booking and Scheduling
- Setting location
- Edit profile
- Payment Gateway
- cancel appointment

Service partner will have the following functionalities

- Registration and Login
- Service Management
- view the location of user
- Payment Managemet
- Edit profile

2.ARCHITECTURE



3. TECHNOLOGY

3.1 FRONT END TOOLS:

- **LottieFiles**(Lottie is an open-source animation file format that's tiny, high quality,scriptable,interactive, and can be manipulated at runtime. The top 500 apps on the App store now use Lottie to engage users and enhance conversions.
- **XML**:XML tags define the data and used to store and organize data. It's easily scalable and simple to develop. In Android, the XML is used to implement UI-related data, and it's a lightweight markup language that doesn't make layout heavy. XML only contains tags, while implementing they need to be just invoked favorite messaging apps and let them serve your customers continuously.

3.2 BACK END TOOLS

Java: High-level, Object-oriented Programming Language

- Widely used for developing various software applications
- Known for platform independence and versatility
- Popular for server-side development and back-end systems

Google Auth: Secure User Authentication

- Offers authentication services for web and mobile apps
- Simplifies user authentication and management

Google Maps API: Location and Mapping Services

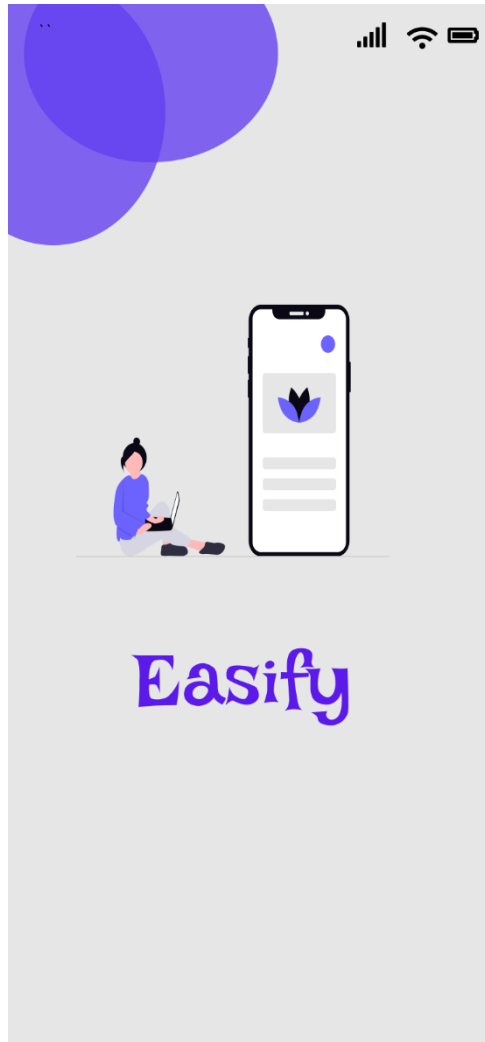
- Provides mapping and location-related functionalities
- Enables integration of maps and location services into applications

3.3 DATABASE

Firebase : Firebase Realtime Database is a cloud-based database that stores and syncs data in real time across multiple devices. It's part of Firebase, a Backend-as-a-Service (BaaS) platform that offers hosted backend services for app development.

4.FIREFRAMES

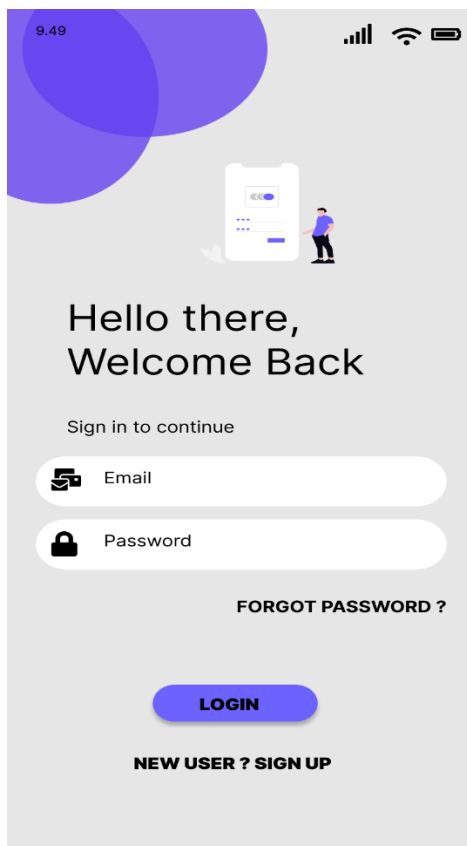
Splash screen



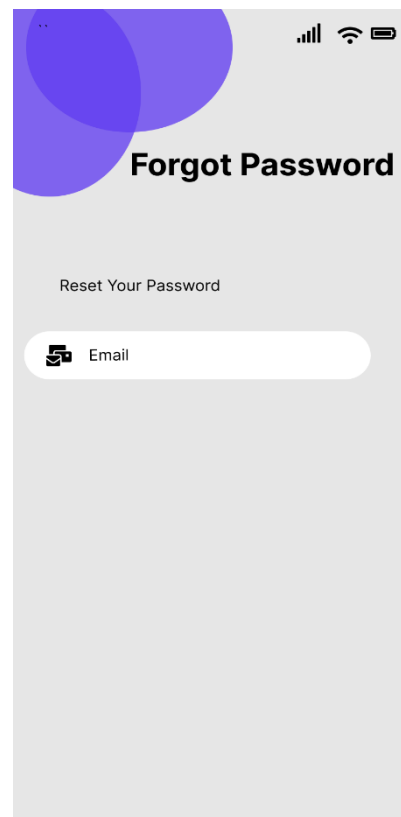
Registration screen

The registration screen has a light gray background with two large, overlapping purple circles in the top left corner. At the top, the time 9:49 is displayed. Below the circles, there is an illustration of a person standing next to a large smartphone that displays a registration form. The text "Welcome," is centered in a large, black, sans-serif font. Below it, the text "Sign up to Start Your Journey" is centered in a smaller, black, sans-serif font. There are four input fields, each with a white background and a rounded rectangle shape. The first field is labeled "Full Name" with a person icon. The second field is labeled "Email" with an envelope icon. The third field is labeled "Phone No" with a phone icon. The fourth field is labeled "Password" with a lock icon. At the bottom, there is a blue button with the word "REGISTER" in white, uppercase letters.

Login Screen



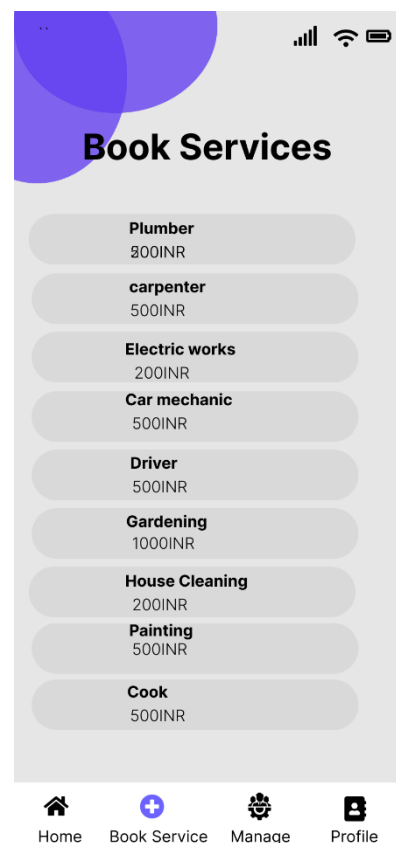
Forgot password




Home screen






Service listing



Book service




PLUMBING

Service required on

SELECT DATE AND TIME





Service required at

Write Address

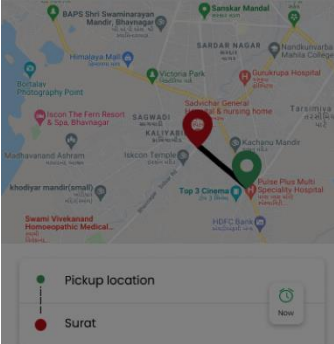
 Select Current Location

SERVICE INFORMATION

Enter Here

 Home
  Book Service
  Manage
  Profile

Scheduling



Pickup location

Surat

Now

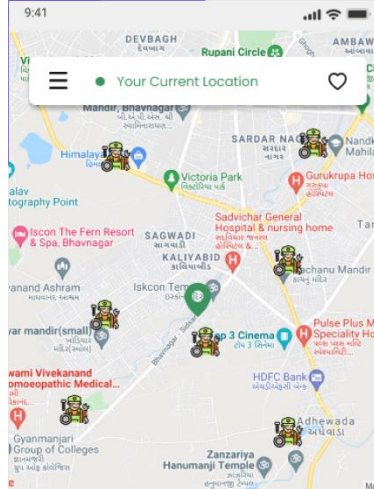
Schedule your booking

4 feb, 01:00 pm

Thu 3 feb	12	45	am
Fri 4 feb	1	00	pm
Sat 5 feb	2	15	

Confirm

setting location



9:41

DEVBAGH Rupani Circle AMBAWADI

manor, bhavnagar

SARDAR NAGAR

Nandkumar Mahila C

Gurukrupa Hospi

Sadvi Char General Hospital & nursing home

SAGWADI

KALIYABID

Iskcon Temple

Top 3 Cinema

Pulse Plus Multi Speciality Hosp

HDFC Bank

Adhewada

Zanzariya Hanumanti Temple

Gyanmanjari Group of Colleges

Swami Vivekanand Homoeopathic Medical...


Choose Your Locations




₹ Offer Your Max Budget

Comments And Wishes

FIND

Manage Order



Manage Order

order id : order_id

user name : cust_name

mobile no : mobile_no





Address : address

landmark : landmark

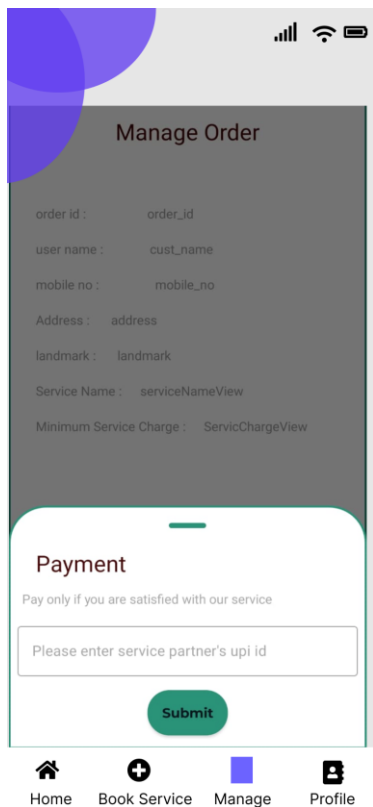
Service Name : serviceNameView

Minimum Service Charge : ServiceChargeView

pay service fee **Cancel Appointment**

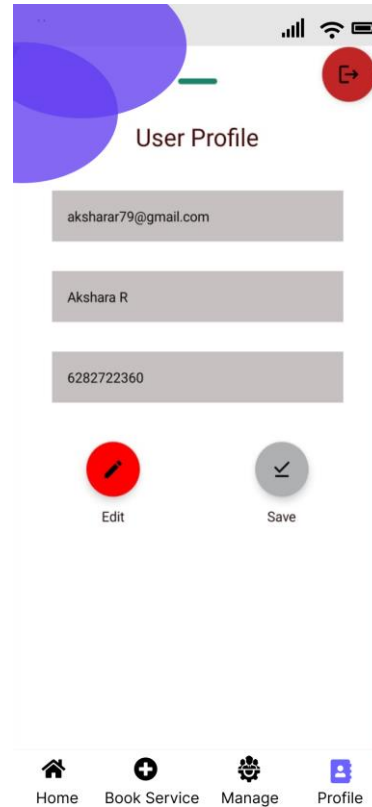
 Home
  Book Service
  Manage
  Profile

Pay for service



Mobile app mockup for the 'Pay for service' screen. The screen has a light gray background with a purple abstract shape in the top left. At the top, there's a status bar with signal, Wi-Fi, and battery icons. Below it, a dark gray header contains the text 'Manage Order'. The main content area is a dark gray card with white text listing order details: 'order id : order_id', 'user name : cust_name', 'mobile no : mobile_no', 'Address : address', 'landmark : landmark', 'Service Name : serviceNameView', and 'Minimum Service Charge : ServiceChargeView'. Below this card is a white rounded rectangle with a green border. It contains the title 'Payment', a subtitle 'Pay only if you are satisfied with our service', a text input field with the placeholder 'Please enter service partner's upi id', and a green 'Submit' button. At the bottom is a navigation bar with four icons: Home, Book Service, Manage, and Profile.

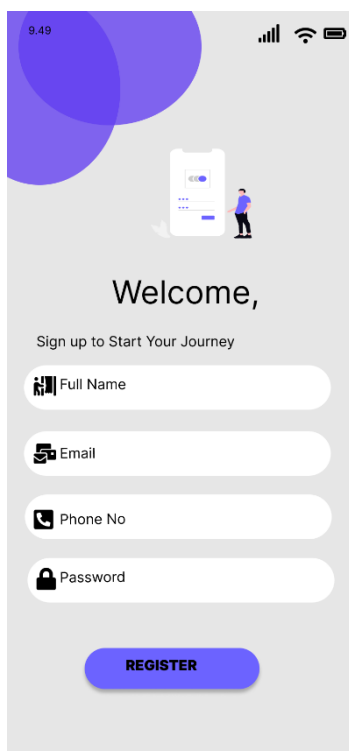
Edit profile



Mobile app mockup for the 'Edit profile' screen. The screen has a light gray background with a purple abstract shape in the top left. At the top, there's a status bar with signal, Wi-Fi, and battery icons. Below it, a dark gray header contains the text 'User Profile'. The main content area is a white card with three gray input fields containing the text 'aksharar79@gmail.com', 'Akshara R', and '6282722360'. Below these fields are two circular buttons: a red one with a pencil icon labeled 'Edit' and a gray one with a checkmark icon labeled 'Save'. At the bottom is a navigation bar with four icons: Home, Book Service, Manage, and Profile.

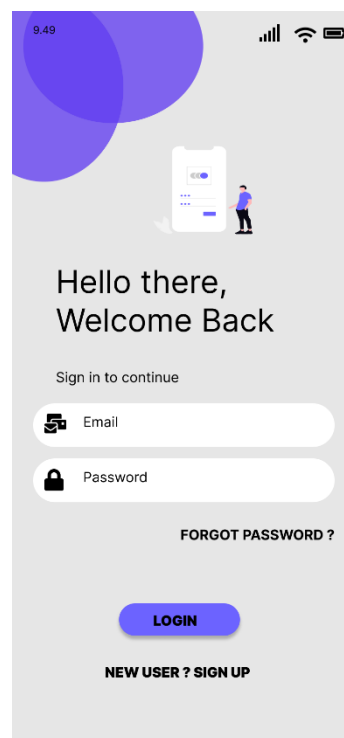
Service parner

Registration



Mobile app mockup for the Registration screen. The screen has a light gray background with a purple abstract shape in the top left. At the top, there's a status bar with the time '9:49', signal, Wi-Fi, and battery icons. Below it, there's an illustration of a person standing next to a smartphone. The main text is 'Welcome,' followed by the subtitle 'Sign up to Start Your Journey'. There are four input fields with icons and labels: 'Full Name', 'Email', 'Phone No', and 'Password'. At the bottom is a blue rounded button labeled 'REGISTER'.

Login

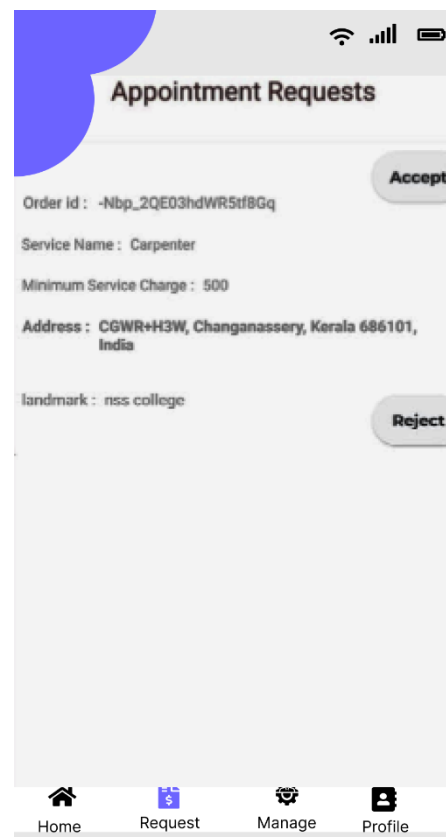


Mobile app mockup for the Login screen. The screen has a light gray background with a purple abstract shape in the top left. At the top, there's a status bar with the time '9:49', signal, Wi-Fi, and battery icons. Below it, there's an illustration of a person standing next to a smartphone. The main text is 'Hello there, Welcome Back' followed by the subtitle 'Sign in to continue'. There are two input fields with icons and labels: 'Email' and 'Password'. Below these fields is a link labeled 'FORGOT PASSWORD ?'. At the bottom is a blue rounded button labeled 'LOGIN' and a text link 'NEW USER ? SIGN UP'.

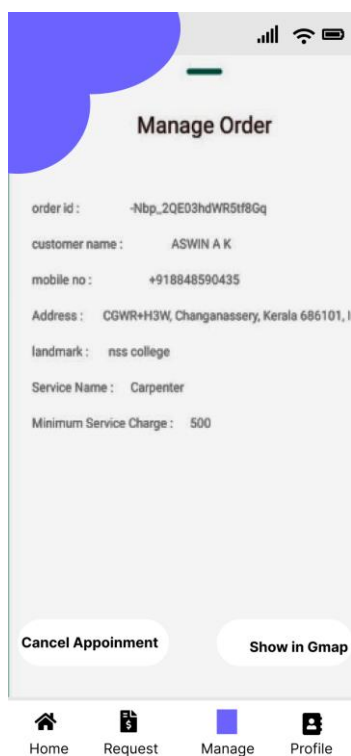
Home



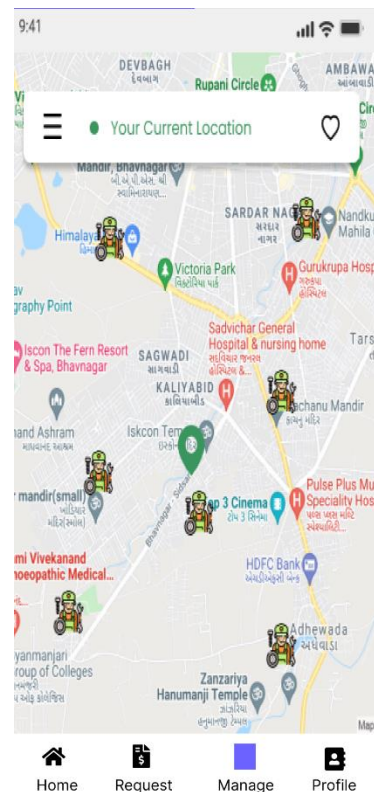
manage request



Manage Order



viewing location



Edit Profile

raghir79@gmail.com

Raghi

6282722360

Edit Save

Home Request Manage Profile

P

Prototype link : <https://www.figma.com/proto/JyOKtRmuv6t3qko5QJRZa2/easify?node-id=0-1&t=UytBtasuZ47TKyNV-1>

5.API

1.User Registration API

- URL: /api/v1/auth/register
- Method: POST
- Input:Email,Password,Name,phone number

Body

```
{
  "email": "user@example.com",
  "password": "password123",
  "name": "John Doe",
  "phone number":9654234587
}
```

- **Output:**

- **Success:**

```
{
  "message": "User registered successfully",
  "userId": "12345"
}
```

- **Failure:**

```
{
  "error": "Email already exists"
}
```

2. User Login API

- URL: /api/v1/auth/login
- Method: POST
- Input:Email,Password

Body:

```
{
  "email":
    "user@example.com",
  "password": "password123"
}
```

- **Output:**
 - **Success:**

```
{  
  "message": "Login successful",  
}
```
 - **Failure:**

```
{  
  "error": "Invalid credentials"  
}
```

3.Forgot Password API

Step 1: Request Password Reset

- **Endpoint:** /api/v1/auth/forgot-password
- **Method:** POST
- **Headers:**
 - Content-Type: application/json

Request Body:

```
{  
  
  "email": "user@example.com"  
}
```

Response:

- **Success:**

```
{  
  
  "status": "success",  
  
  "message": "Password reset link has been sent to your email address."  
}
```

- **Failure:**

```
{  
  
  "status": "error",  
}
```

```
"message": "Email address not found in our system."
}
```

4. Fetch Available Services API

- URL: /api/v1/services
- Method: GET
- Input:None
 - Output: Success:

```
{
  "services": [
    {
      "id": "1",
      "name": "Plumbing",
      "description": "Professional plumbing services"
    },
    {
      "id": "2",
      "name": "Electrician",
      "description": "Electrical services for home and
office"
    }
  ]
}
```

- Failure:

```
{
  "error": "Unable to fetch services"
}
```

5. Search Services API

- URL: /api/v1/services/search
- Method: GET
- Input:Query parameter: q (search term)
- Body: None
- Output:
 - Success:

```
{
  "results": [
    {
      "id": "3",
      "name": "Carpentry",
      "description": "Custom furniture and repairs"
    }
  ]
}
```

- Failure:

```
{
  "error": "No matching services found"
}
```

6. Book Appointment API

- URL: /api/v1/appointments
- Method: POST
- Input:
 - User token (JWT)
 - Service ID
 - Location and landmark
 - Appointment time
- Body:

```
{
  "serviceId": "2",
  "location": "123 Main Street",
  "landmark": "Near City Park",
  "appointmentTime": "2025-01-21T10:00:00Z"
}
```

- Output:
 - Success:

```
{
  "message": "Appointment booked successfully",
  "appointmentId": "56789"
}
```
 - Failure:

```
{
```

```
"error": "Service provider unavailable at selected time"
}
```

7.API for Setting Location

- **URL:** /api/v1/user/set-location
- **Method:** POST
- **Input (Body):**

```
{
  "latitude": 12.971598,
  "longitude": 77.594566,
  "address": "Electronic City,
Bangalore",
  "landmark": "Near Tech Park"
}
```

- **Response:**

- **Success:**

```
{
  "status": "success",
  "message": "Location updated successfully."
}
```

- **Failure:**

```
{
  "error": "Location could not be updated. Please try again."
}
```

8.Initiate Payment

Endpoint: /api/v1/payment/initiate

- **Method:** POST
- **Description:** Initiates the payment process and generates a payment order.

Request Body:

```
{
  "user_id": "12345",
  "service_id": "67890",
  "amount": 1000.00,
  "currency": "INR",
  "payment_method": "card" // options: 'card', 'wallet',
'UPI'
}
```

Response:

- **Success:**

```
{
  "status": "success",
  "message": "Payment initiated successfully.",
  "order_id": "ORDER123456",
  "payment_gateway_url": "https://paymentgateway.com/pay?order_id=ORDER123456"
}
```

- **Failure:**

```
{
  "status": "error",
  "message": "Failed to initiate payment. Please try again."
}
```

9.Cancel Appointment API

- **URL:** /api/v1/appointments/{appointmentId}

- **Method:** DELETE

- **Input:**

- User token (JWT)
- Appointment ID (in URL)

- **Body:** None

- **Output:**

- **Success:**

```
{
  "message": "Appointment cancelled successfully"
}
```

- **Failure:**

```
json
CopyEdit
{
  "error": "Appointment not found or already cancelled"
}
```

10. Edit Profile API

- **URL:** /api/v1/user/profile

- **Method:** PUT

- **Input:**

- User token (JWT)
- Updated email or phone number
- **Body:**

```
{  
  "email":  
    "newemail@example.com",  
  "phone": "+1234567890"  
}
```

- **Output:**

- **Success:**

```
{  
  "message": "Profile updated successfully"  
}
```

- **Failure:**

```
{  
  "error": "Invalid input or email already in use"  
}
```

11. Service Partner Registration API

- **URL:** /api/v1/partner/register
- **Method:** POST
- **Input:** Email, Phone number, Password, Name
- **Body:**

```
{  
  "email":  
    "partner@example.com",  
  "password": "securepassword",  
  "phone": "+1234567890",  
  "name": "John Smith"  
}
```

- **Output:**

- **Success:**

```
{  
  "message": "Partner registered successfully",  
  "partnerId": "12345"  
}
```

```
}
```

- **Failure:**

```
"error": "Email or phone number already exists"
}
```

12. Service Partner Login API

- **URL:** /api/v1/partner/login
- **Method:** POST
- **Input:**
 - Email
 - Password
- **Body:**

```
{
  "email":
    "partner@example.com",
  "password":
    "securepassword"
}
```

- **Output:**
 - Success:

```
{
  "message": "Login successful",
  "token": "jwt_token_here"
}
```
 - **Failure:**

```
{
  "error": "Invalid credentials"
}
```

13. Manage Bookings API

- **URL:** /api/v1/partner/bookings
- **Method:** GET (fetch bookings), PUT (update status)
- **Input:**
 - Partner token (JWT)
 - Booking details: appointment ID, status (for PUT)
- **Body:**

```
{
  "appointmentId": "56789",
  "status": "Accepted" // Accepted, Rejected,
  Completed
}
```

- **Output**

- **Success (PUT):**

```
{
  "message": "Booking status updated successfully"
}
```

- **Failure:**

```
{
  "error": "Failed to update booking status"
}
```

14.View Customer Location API

- **URL:** /api/v1/partner/booking/{appointmentId}/location
- **Method:** GET
- **Input:**
 - Partner token (JWT)
 - Appointment ID (in URL)
- **Body:** None
- **Output:**
 - **Success:**

```
"location": {
  "address": "123 Main Street",
  "landmark": "Near City Park",
  "latitude": 37.7749,
  "longitude": -122.4194
},
"googleMapsUrl": "https://www.google.com/maps/search/?api=1&query=37.7749,-
122.4194",
"user": {
  "name": "Jane Doe",
  "phone": "+9876543210"
}
}
```

- **Failure:**

```
{
  "error": "Booking not found or unauthorized"
}
```

15. Edit Partner Profile API

- **URL:** /api/v1/partner/profile
- **Method:** PUT
- **Input:**
 - Partner token (JWT)
 - Updated email, phone number, or name
- **Body:**

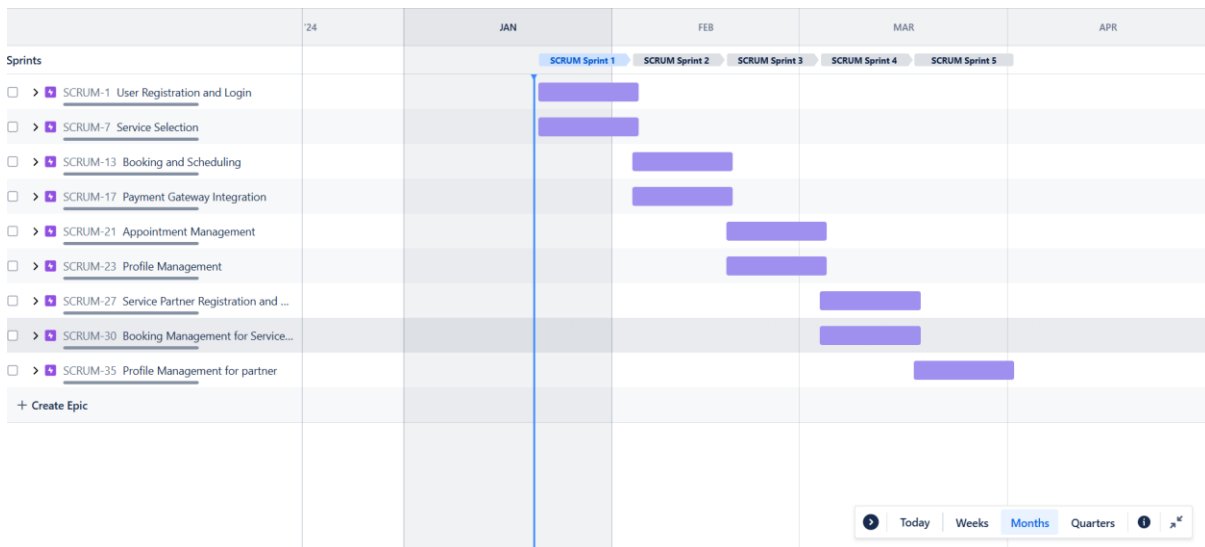
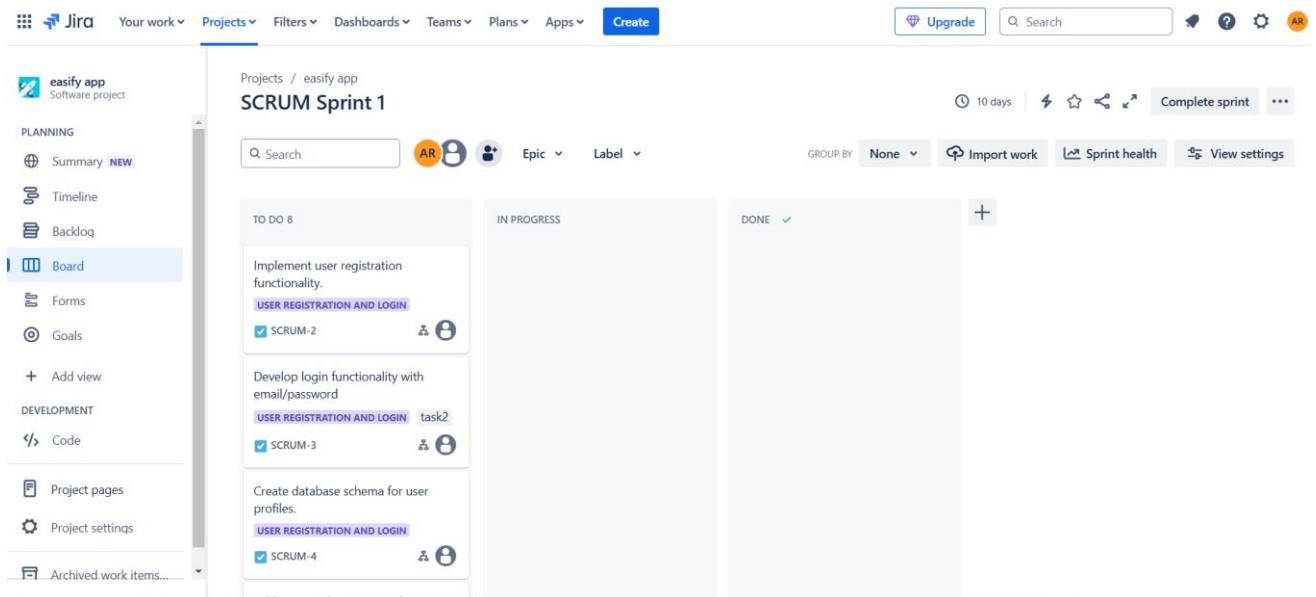
```
{
  "email":
    "newemail@example.com",
  "phone": "+1234567890",
  "name": "New Name"
}
```

- **Output:**
 - **Success:**

```
{
  "message": "Profile updated successfully"
}
```
 - **Failure:**

```
{
  "error": "Invalid input or email already in use"
}
```

6.JIRA



7.GIT

The link of newly created Github repository where I can push the code

<https://github.com/Akhara08/easifyapp.git>

MILESTONE

(21-1-2025) To (4-2-2025)

- Demo the login functionality by accepting a username and password.
- Demo user registration and profile creation.
- Enable service selection from the list of services

(4-1-2025) to (18-2-2025)

- Demo booking and scheduling functionality with location input.
- We will demo the Integrattion of a payment gateway with dummy payment

(18-2-2025) to (4-3-2025)

- Demo the option to cancel appointments and update the status.
- We will demo the profile editing for phone number and email updates.

(4-3-2025) to (18-3-2025)

- Demo the login functionality by accepting a username and password for service partners.
- demo service partner registration and profile creation.
- Demo booking management, allowing partners to view, accept, or reject appointments.
- Demo the functionality to view the customer's location for accepted bookings.

(18-3-2025) to (25-3-2025)

- Demo the profile editing for service partners to update their phone number and email.