

DBMS PROJECT REPORT

1. Problem Statement: HOME APPLIANCE SERVICE MANAGEMENT SYSTEM

TEAM:

Ashwin (PES1UG23CS117)

Ashish Kumar B (PES1UG23CS114)

2. Description about the Statement:

Managing home appliance repairs involves multiple challenges: customers struggle to find reliable technicians with the right expertise, service centers face difficulties in efficiently assigning technicians based on skills and availability, and there's limited transparency in tracking service requests from booking to completion.

Traditional manual coordination leads to delayed responses, suboptimal technician utilization, and poor customer experience due to lack of real-time status updates and payment tracking. The system needs to automate technician assignment based on appliance type, location, workload, and performance ratings while providing separate portals for customers and technicians to manage their respective workflows efficiently.

3. User Requirement Specification:

Purpose of the project:

The Home Appliance Service Management System aims to streamline the process of booking, tracking, and managing home appliance repair services. It addresses the inefficiencies in traditional manual coordination by providing an automated platform that connects customers needing appliance repairs with qualified technicians. The system ensures optimal technician assignment based on skills, location, availability, and performance ratings while maintaining transparency throughout the service lifecycle from booking to payment completion.

Scope of the project:

The system encompasses two primary user interfaces: a customer portal for booking services, tracking requests, managing appliances, and rating technicians; and a technician portal for viewing assigned jobs, updating service status, generating invoices, and managing professional profiles. The backend implements role-based database access control with separate user credentials for customers and technicians, ensuring data security and proper privilege management. The system covers complete service request lifecycle management including automated technician assignment, real-time status tracking, invoice generation, payment tracking, and performance rating aggregation.

Detailed Description:

The Home Appliance Service Management System is a full-stack web application built using React.js for the frontend, Node.js with Express.js for the backend API, and MySQL for database management. Customers register with their personal details and address, then add their home appliances (refrigerator, washing machine, AC, microwave, TV) to the system. When booking a service, customers select an appliance, describe the issue, and submit the request. The system uses intelligent SQL triggers to automatically assign the most suitable technician based on matching skills for the appliance type, location proximity (same city as customer), current workload (fewest active jobs), and performance rating (tie-breaker). Technicians receive assigned requests in their portal, can start jobs (changing status to "In Progress"), and finish jobs by creating invoices with cost details (automatically marking status as "Completed"). Customers can then rate the completed service, which updates the technician's average rating in real-time. The system maintains separate database connection pools for customer and technician operations, ensuring each user type only accesses data they're authorized to view or modify. Service centers are pre-configured with locations, and technicians are assigned to specific centers with defined skill sets. The platform supports multi-phone number storage per customer, appliance brand and model tracking, payment status monitoring (Unpaid/Paid), and comprehensive service history retrieval through stored procedures.

Functional Requirements:

FR-1.0: Customer Registration and Authentication

- FR-1.1: The customer must be able to register by providing first name, email, password, date of birth, address, and phone number
- FR-1.2: The system must validate email uniqueness and pincode/phone number format
- FR-1.3: The customer must be able to login using email and password with JWT token generation

FR-2.0: Appliance Management

- FR-2.1: The customer must be able to add appliances by specifying type, brand, and model number
- FR-2.2: The customer must be able to view all registered appliances

FR-3.0: Service Request Management (Customer)

- FR-3.1: The customer must be able to create service requests by selecting an appliance and providing problem description
- FR-3.2: The customer must be able to view all service requests with status (Pending, In Progress, Completed) and assigned technician details
- FR-3.3: The customer must be able to cancel pending service requests
- FR-3.4: The customer must be able to rate completed services on a 1-5 scale

FR-4.0: Customer Profile Management

- FR-4.1: The customer must be able to view and update personal information, address, and phone numbers
- FR-4.2: The customer must be able to change password after verification

FR-5.0: Technician Authentication and Job Management

- FR-5.1: The technician must be able to login using technician ID
- FR-5.2: The technician must be able to view assigned service requests with customer and appliance details
- FR-5.3: The technician must be able to start pending jobs and finish in-progress jobs by creating invoices
- FR-5.4: The technician must be able to mark invoices as paid

FR-6.0: Technician Profile Management

- FR-6.1: The technician must be able to view profile including rating, completed jobs, and active job count
- FR-6.2: The technician must be able to update personal information and upload profile photo
- FR-6.3: The technician must be able to add and remove skills

FR-7.0: Automated Technician Assignment

- FR-7.1: The system must automatically assign technicians based on skill match, location, current workload, and rating
- FR-7.2: The system must automatically recalculate technician ratings when customers submit service ratings

FR-8.0: Access Control and Security

- FR-8.1: The system must implement separate database credentials for customer and technician operations with appropriate table-level privileges
- FR-8.2: The system must store passwords as SHA-256 hashed values and validate JWT tokens for all API requests

4. List of Softwares/Tools/Programming languages used:

Frontend

- **React.js** - JavaScript library for building user interfaces
- **React Router** - Client-side routing for navigation
- **Axios** - HTTP client for API requests

- **CSS3** - Styling and responsive design

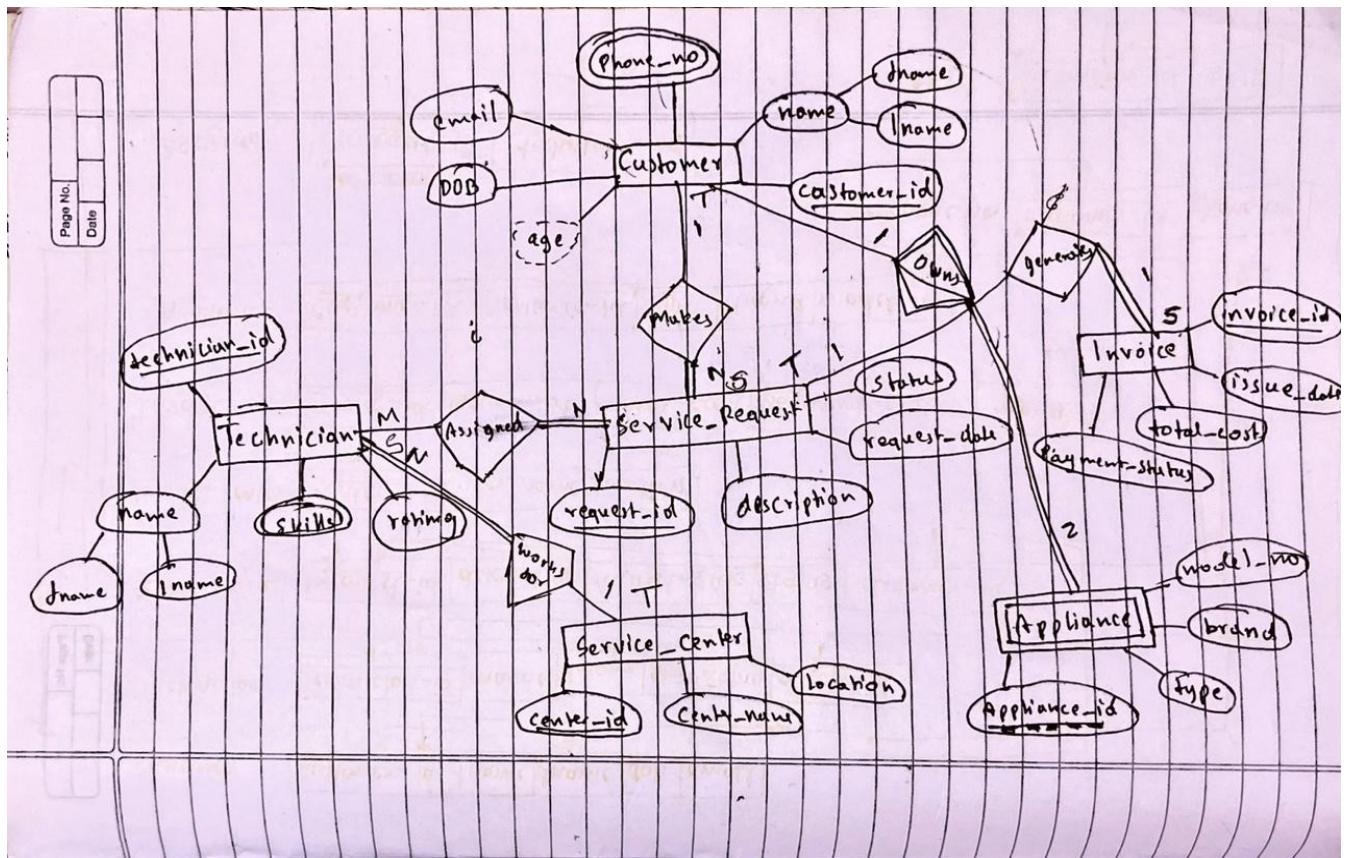
Backend

- **Node.js** - JavaScript runtime environment
- **Express.js** - Web application framework for REST API
- **JWT (jsonwebtoken)** - Token-based authentication
- **Multer** - Middleware for file uploads
- **crypto** - Password hashing (SHA-256)
- **dotenv** - Environment variable management
- **mysql2** - MySQL database driver with promise support

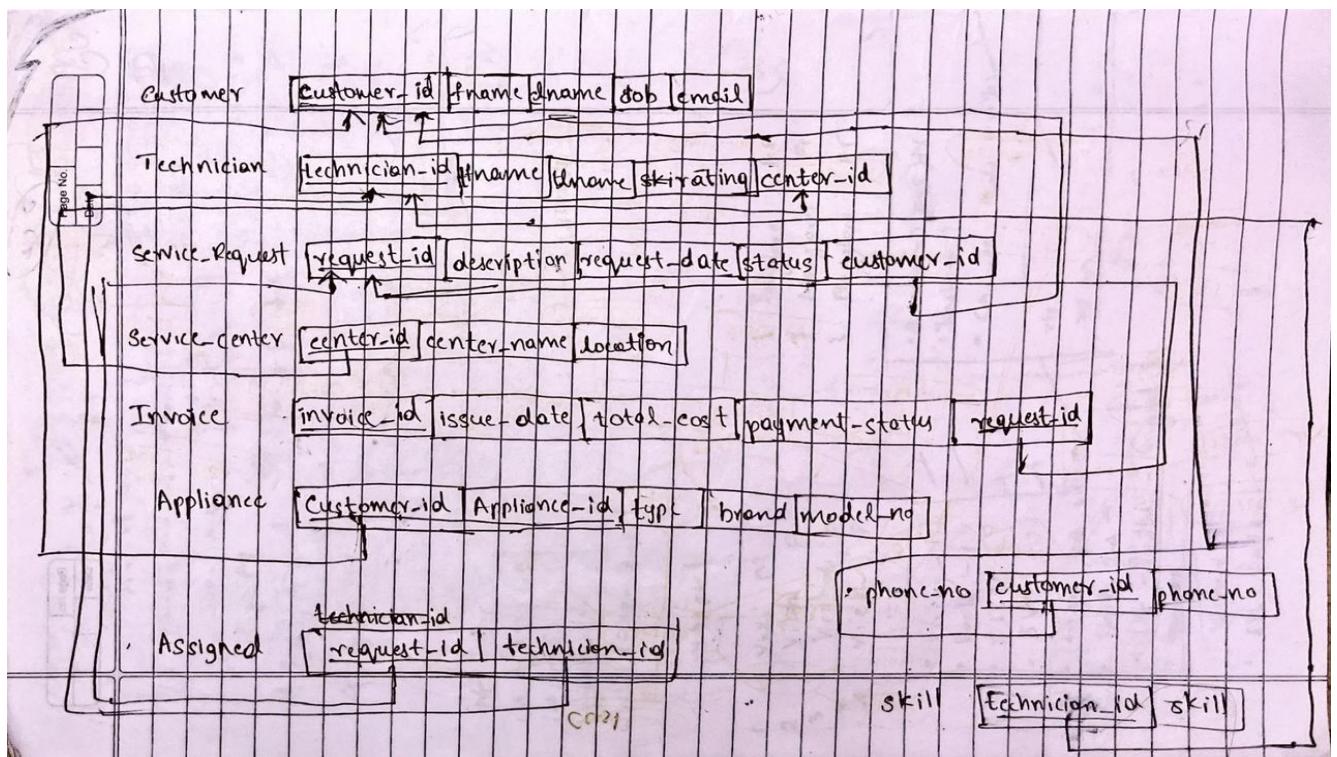
Database

- **MySQL 8.0** - Relational database management system
- **SQL** - Database queries, triggers, stored procedures, and functions

5. ER Diagram:



6. Relational Schema:



7. DDL Commands:

```
CREATE DATABASE IF NOT EXISTS home_service_db;
```

```
USE home_service_db;
```

```

CREATE TABLE Service_Center (
    center_id INT AUTO_INCREMENT PRIMARY KEY,
    center_name VARCHAR(100) NOT NULL,
    location VARCHAR(100) NOT NULL
);
    
```

```

CREATE TABLE Customer (
    customer_id INT AUTO_INCREMENT PRIMARY KEY,
    fname VARCHAR(50) NOT NULL,
    lname VARCHAR(50),
    dob DATE,
    );
    
```

```
email VARCHAR(100),  
password_hash VARCHAR(255) NOT NULL,  
address_line1 VARCHAR(100) NOT NULL,  
landmark VARCHAR(100),  
stage VARCHAR(50),  
city VARCHAR(50) NOT NULL,  
pincode CHAR(6) NOT NULL CHECK (pincode REGEXP '^[1-9][0-9]{5}$')  
);
```

```
CREATE TABLE Technician (  
technician_id INT AUTO_INCREMENT PRIMARY KEY,  
fname VARCHAR(50) NOT NULL,  
lname VARCHAR(50),  
phone_no VARCHAR(15),  
rating DECIMAL(2,1) CHECK (rating BETWEEN 0 AND 5),  
photo LONGBLOB,  
center_id INT,  
FOREIGN KEY (center_id) REFERENCES Service_Center(center_id)  
    ON DELETE CASCADE  
    ON UPDATE CASCADE  
);
```

```
CREATE TABLE Appliance (  
appliance_id INT,  
customer_id INT,  
type VARCHAR(50) NOT NULL,  
brand VARCHAR(50),  
model_no VARCHAR(50),  
PRIMARY KEY (appliance_id, customer_id),  
FOREIGN KEY (customer_id) REFERENCES Customer(customer_id)  
    ON DELETE CASCADE
```

```

    ON UPDATE CASCADE
);

CREATE TABLE Service_Request (
    request_id INT AUTO_INCREMENT PRIMARY KEY,
    description VARCHAR(255),
    request_date DATE,
    status VARCHAR(20) DEFAULT 'Pending'
        CHECK (status IN ('Pending', 'In Progress', 'Completed', 'Cancelled')),
    rating DECIMAL(2,1) CHECK (rating BETWEEN 1 AND 5),
    customer_id INT NOT NULL,
    technician_id INT,
    appliance_id INT,
    FOREIGN KEY (appliance_id, customer_id)
        REFERENCES Appliance(appliance_id, customer_id)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    FOREIGN KEY (customer_id) REFERENCES Customer(customer_id)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    FOREIGN KEY (technician_id) REFERENCES Technician(technician_id)
        ON DELETE SET NULL
        ON UPDATE CASCADE
);

```

```

CREATE TABLE Invoice (
    invoice_id INT AUTO_INCREMENT PRIMARY KEY,
    request_id INT NOT NULL,
    issue_date DATE,
    total_cost DECIMAL(10,2) NOT NULL CHECK (total_cost >= 0),
    payment_status VARCHAR(20) DEFAULT 'Unpaid'

```

```
    CHECK (payment_status IN ('Unpaid', 'Paid', 'Pending')),  
    FOREIGN KEY (request_id) REFERENCES Service_Request(request_id)  
        ON DELETE CASCADE  
        ON UPDATE CASCADE  
);
```

```
CREATE TABLE Skill (  
    technician_id INT NOT NULL,  
    skill VARCHAR(50) NOT NULL,  
    PRIMARY KEY (technician_id, skill),  
    FOREIGN KEY (technician_id) REFERENCES Technician(technician_id)  
        ON DELETE CASCADE  
        ON UPDATE CASCADE  
);
```

```
CREATE TABLE Phone_no (  
    customer_id INT NOT NULL,  
    phone_no VARCHAR(15) NOT NULL  
        CHECK (phone_no REGEXP '^[0-9]+$',  
    PRIMARY KEY (phone_no),  
    FOREIGN KEY (customer_id) REFERENCES Customer(customer_id)  
        ON DELETE CASCADE  
        ON UPDATE CASCADE  
);
```

```
DELIMITER $$
```

```
CREATE TRIGGER after_invoice_insert_complete  
AFTER INSERT ON Invoice  
FOR EACH ROW  
BEGIN
```

```
UPDATE Service_Request  
SET status='Completed'  
WHERE request_id = NEW.request_id;  
END$$
```

```
CREATE TRIGGER after_service_rating_update  
AFTER UPDATE ON Service_Request  
FOR EACH ROW  
BEGIN  
IF NEW.rating IS NOT NULL AND OLD.rating IS NULL AND NEW.technician_id IS NOT NULL THEN  
    UPDATE Technician  
    SET rating = (  
        SELECT ROUND(AVG(rating), 1)  
        FROM Service_Request  
        WHERE technician_id = NEW.technician_id  
        AND rating IS NOT NULL  
    )  
    WHERE technician_id = NEW.technician_id;  
END IF;  
END$$
```

```
CREATE TRIGGER before_sr_insert_assign  
BEFORE INSERT ON Service_Request  
FOR EACH ROW  
BEGIN  
DECLARE v_type VARCHAR(100);  
DECLARE v_best_tech INT;  
  
IF NEW.appliance_id IS NOT NULL AND NEW.customer_id IS NOT NULL AND NEW.technician_id IS NULL THEN  
    SELECT type INTO v_type
```

```
FROM Appliance  
WHERE appliance_id = NEW.appliance_id AND customer_id = NEW.customer_id LIMIT 1;
```

```
IF v_type IS NOT NULL THEN  
    SELECT t.technician_id INTO v_best_tech  
    FROM Technician t  
    INNER JOIN Skill s ON s.technician_id = t.technician_id AND s.skill = v_type  
    INNER JOIN Service_Center sc ON sc.center_id = t.center_id  
    INNER JOIN Customer cu ON cu.customer_id = NEW.customer_id  
    LEFT JOIN Service_Request sr ON sr.technician_id = t.technician_id  
        AND sr.status IN ('Pending','In Progress')  
    WHERE sc.location = cu.city  
    GROUP BY t.technician_id  
    ORDER BY COUNT(sr.request_id) ASC, t.rating DESC  
    LIMIT 1;
```

```
IF v_best_tech IS NOT NULL THEN  
    SET NEW.technician_id = v_best_tech;  
END IF;  
END IF;  
END IF;  
END$$
```

```
CREATE TRIGGER before_sr_update_reassign  
BEFORE UPDATE ON Service_Request  
FOR EACH ROW  
BEGIN  
    DECLARE v_type VARCHAR(100);  
    DECLARE v_best_tech INT;  
  
    IF OLD.technician_id IS NOT NULL AND NEW.technician_id IS NULL AND NEW.status = 'Pending' THEN
```

```

SELECT type INTO v_type
FROM Appliance
WHERE appliance_id = NEW.appliance_id AND customer_id = NEW.customer_id LIMIT 1;

IF v_type IS NOT NULL THEN
    SELECT t.technician_id INTO v_best_tech
    FROM Technician t
    INNER JOIN Skill s ON s.technician_id = t.technician_id AND s.skill = v_type
    INNER JOIN Service_Center sc ON sc.center_id = t.center_id
    INNER JOIN Customer cu ON cu.customer_id = NEW.customer_id
    LEFT JOIN Service_Request sr ON sr.technician_id = t.technician_id
        AND sr.status IN ('Pending','In Progress')
    WHERE sc.location = cu.city
    GROUP BY t.technician_id
    ORDER BY COUNT(sr.request_id) ASC, t.rating DESC
    LIMIT 1;

    IF v_best_tech IS NOT NULL THEN
        SET NEW.technician_id = v_best_tech;
    END IF;
END IF;
END IF;

END$$

CREATE FUNCTION get_average_rating(p_center_id INT)
RETURNS DECIMAL(3,2)
DETERMINISTIC
BEGIN
    DECLARE avg_rating DECIMAL(3,2) DEFAULT 0;

```

```
SELECT AVG(rating) INTO avg_rating FROM Technician WHERE center_id=p_center_id;  
RETURN IFNULL(avg_rating,0);  
END$$
```

```
CREATE PROCEDURE get_customer_service_summary(IN cust_id INT)  
BEGIN  
    SELECT  
        sr.request_id,  
        sr.description,  
        sr.request_date,  
        sr.status,  
        sr.rating AS service_rating,  
        a.type AS appliance_type,  
        a.brand AS appliance_brand,  
        a.model_no AS appliance_model,  
        t.technician_id,  
        CONCAT(t.fname, ' ', IFNULL(t.lname, '')) AS technician_name,  
        t.phone_no AS technician_phone,  
        t.photo AS technician_photo,  
        i.total_cost,  
        i.payment_status  
    FROM Service_Request sr  
    JOIN Appliance a ON sr.appliance_id = a.appliance_id AND sr.customer_id = a.customer_id  
    LEFT JOIN Technician t ON sr.technician_id = t.technician_id  
    LEFT JOIN Invoice i ON sr.request_id = i.request_id  
    WHERE sr.customer_id = cust_id  
    ORDER BY sr.request_date DESC, sr.request_id DESC;  
END$$
```

```
DELIMITER ;
```

```
CREATE USER 'customer_user'@'localhost' IDENTIFIED BY 'customer_pass123';

GRANT ALL PRIVILEGES ON home_service_db.Customer TO 'customer_user'@'localhost';
GRANT ALL PRIVILEGES ON home_service_db.Phone_no TO 'customer_user'@'localhost';
GRANT ALL PRIVILEGES ON home_service_db.Service_Request TO 'customer_user'@'localhost';
GRANT ALL PRIVILEGES ON home_service_db.Appliance TO 'customer_user'@'localhost';

GRANT SELECT ON home_service_db.Technician TO 'customer_user'@'localhost';
GRANT SELECT ON home_service_db.Service_Center TO 'customer_user'@'localhost';
GRANT SELECT ON home_service_db.Skill TO 'customer_user'@'localhost';
GRANT SELECT ON home_service_db.Invoice TO 'customer_user'@'localhost';

GRANT EXECUTE ON PROCEDURE home_service_db.get_customer_service_summary TO
'customer_user'@'localhost';

CREATE USER 'technician_user'@'localhost' IDENTIFIED BY 'technician_pass123';

GRANT ALL PRIVILEGES ON home_service_db.Technician TO 'technician_user'@'localhost';
GRANT ALL PRIVILEGES ON home_service_db.Skill TO 'technician_user'@'localhost';
GRANT ALL PRIVILEGES ON home_service_db.Service_Request TO 'technician_user'@'localhost';
GRANT ALL PRIVILEGES ON home_service_db.Invoice TO 'technician_user'@'localhost';

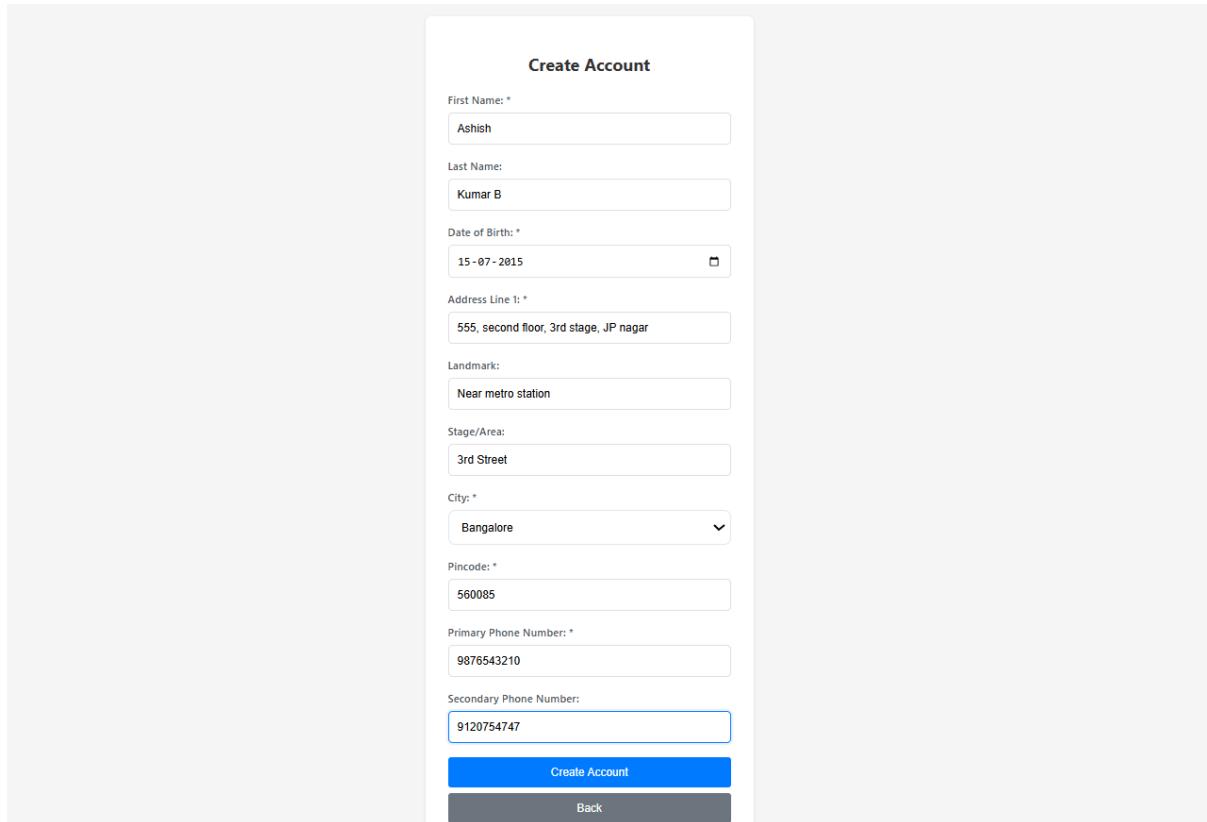
GRANT SELECT ON home_service_db.Customer TO 'technician_user'@'localhost';
GRANT SELECT ON home_service_db.Phone_no TO 'technician_user'@'localhost';
GRANT SELECT ON home_service_db.Appliance TO 'technician_user'@'localhost';
GRANT SELECT ON home_service_db.Service_Center TO 'technician_user'@'localhost';

GRANT EXECUTE ON FUNCTION home_service_db.get_average_rating TO 'technician_user'@'localhost';
```

8. CRUD Operation Screenshots:

CREATE Operations:

1. Customer Registration:

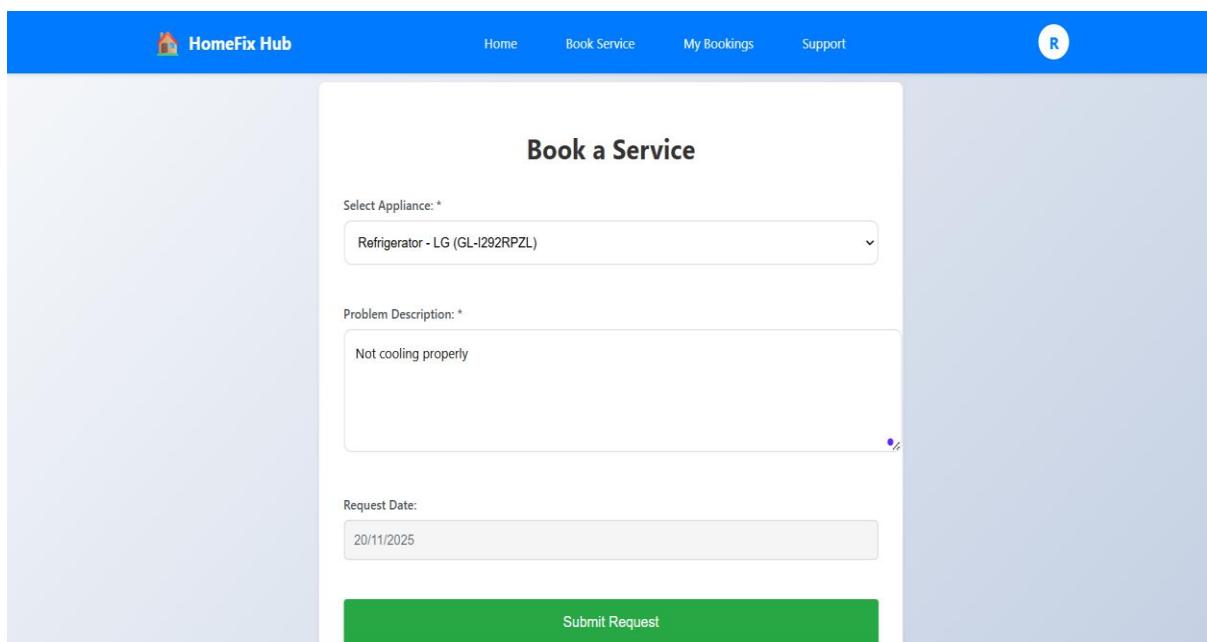


The screenshot shows a 'Create Account' form for customer registration. The fields and their values are as follows:

- First Name: * Ashish
- Last Name: Kumar B
- Date of Birth: * 15-07-2015
- Address Line 1: * 555, second floor, 3rd stage, JP nagar
- Landmark: Near metro station
- Stage/Area: 3rd Street
- City: * Bangalore
- Pincode: * 560085
- Primary Phone Number: * 9876543210
- Secondary Phone Number: 9120754747

At the bottom, there are two buttons: a blue 'Create Account' button and a grey 'Back' button.

2. Create Service Request:



The screenshot shows a 'Book a Service' form. The fields and their values are as follows:

- Select Appliance: * Refrigerator - LG (GL-I292RPZL)
- Problem Description: * Not cooling properly
- Request Date: 20/11/2025

At the bottom, there is a large green 'Submit Request' button.

READ Operations:**3. View Service Requests (Customer):**

The screenshot shows the "My Bookings" section of the HomeFix Hub app. A specific service request for a refrigerator is displayed. The request details are as follows:

- Appliance:** Refrigerator - LG (GL-I292RPZL)
- Problem:** Not cooling properly
- Request Date:** 19/11/2025
- Technician:** Arjun Patel (represented by a placeholder profile picture)
- Contact:** 9876500019

A red button at the bottom right of the card says "Cancel Request".

4. View Assigned Jobs (Technician):

The screenshot shows the "My Service Requests" section of the HomeFix Hub app for a technician. An assigned service request is shown. The request details are as follows:

Customer Information

- Name:** Ravi Kumar
- Phone:** 9800011001, 9800011002
- Address:** #12, 2nd Floor, 5th Cross, Near BTM Bus Stop, BTM Layout, Bangalore, 560076

Service Details

- Appliance:** Refrigerator - LG (GL-I292RPZL)
- Issue:** Not cooling properly
- Request Date:** 19/11/2025

A green button at the bottom right of the card says "Start Job".

UPDATE Operations:

5. Update Service Status:

The screenshot shows the 'My Service Requests' section of the HomeFix Hub application. A specific request, Request #10, is selected. The status of the request is shown as 'In Progress'. The request details are displayed in two sections: 'Customer Information' and 'Service Details'. Under 'Customer Information', the name is Ravi Kumar, phone number is 9800011001, 9800011002, and address is #12, 2nd Floor, 5th Cross, Near BTM Bus Stop, BTM Layout, Bangalore, 560076. Under 'Service Details', the appliance is a Refrigerator - LG (GL-I292RPZL), the issue is 'Not cooling properly', and the request date is 19/11/2025. At the bottom of the request card is a blue button labeled 'Finish Job' with a checkmark icon.

DELETE Operations:

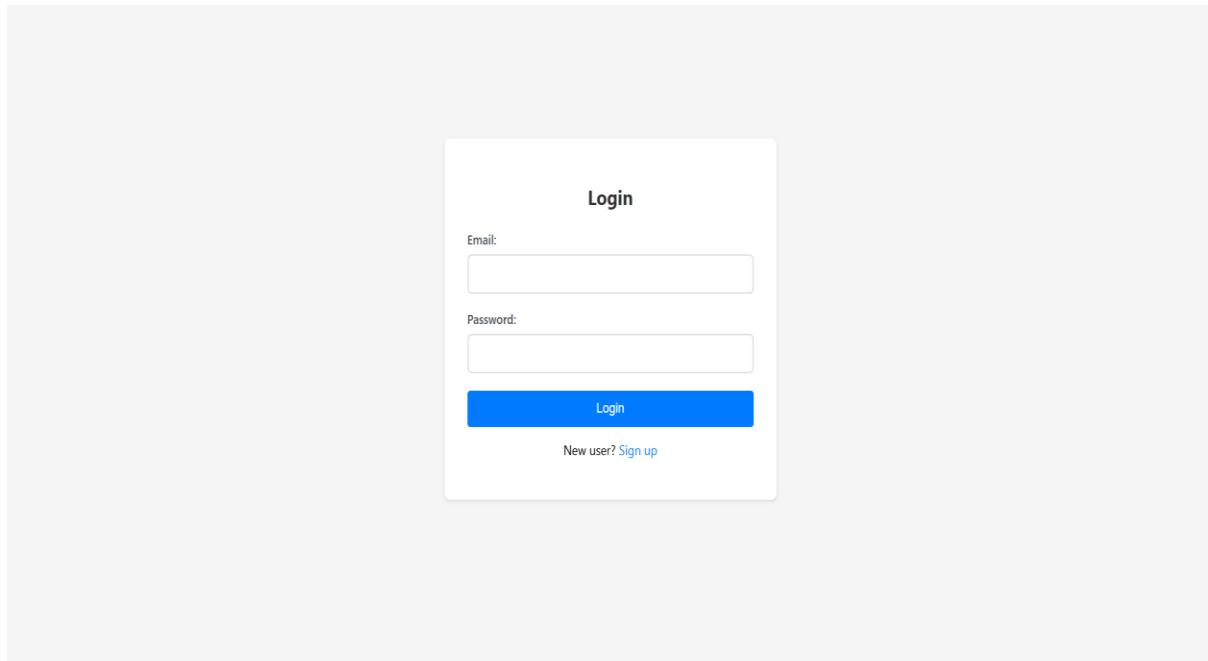
6. Cancel Service Request:

The screenshot shows the 'Refrigerator Service' request page. The request details are identical to the previous update: Appliance: Refrigerator - LG (GL-I292RPZL), Problem: Not cooling properly, Request Date: 19/11/2025. The technician assigned is Arun Singh, and the contact number is 9876500001. A modal dialog box is overlaid on the page, displaying the message 'localhost:3000 says Service request cancelled successfully' with an 'OK' button. The main request card has a yellow 'Pending' status indicator at the top right.

9. Feature Screenshots:

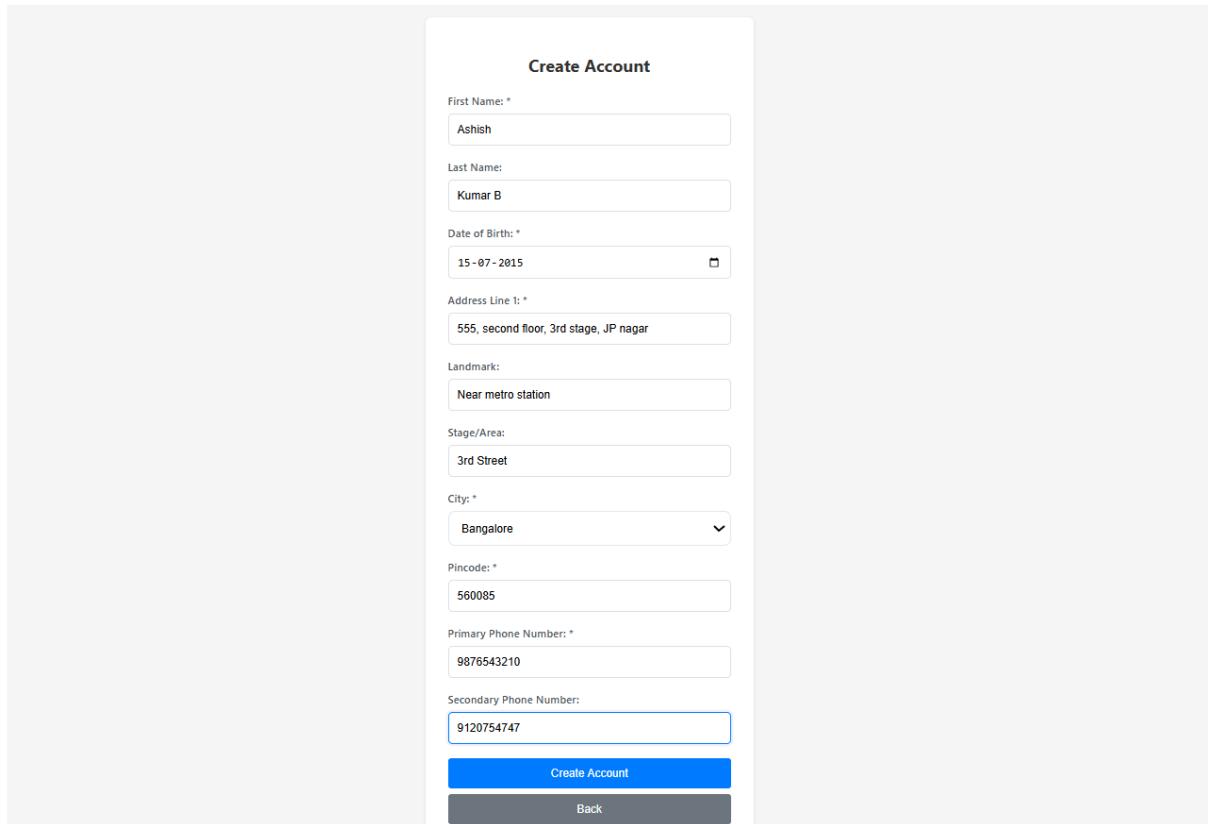
Customer portal:

1. Customer Login:



The screenshot shows a simple login interface titled "Login". It features two input fields: "Email:" and "Password:", both represented by white input boxes with black outlines. Below these fields is a blue rectangular button with the word "Login" in white. At the bottom of the form, there is a small link in blue text that reads "New user? [Sign up](#)". The entire form is centered on a light gray background.

2. Customer Registration:

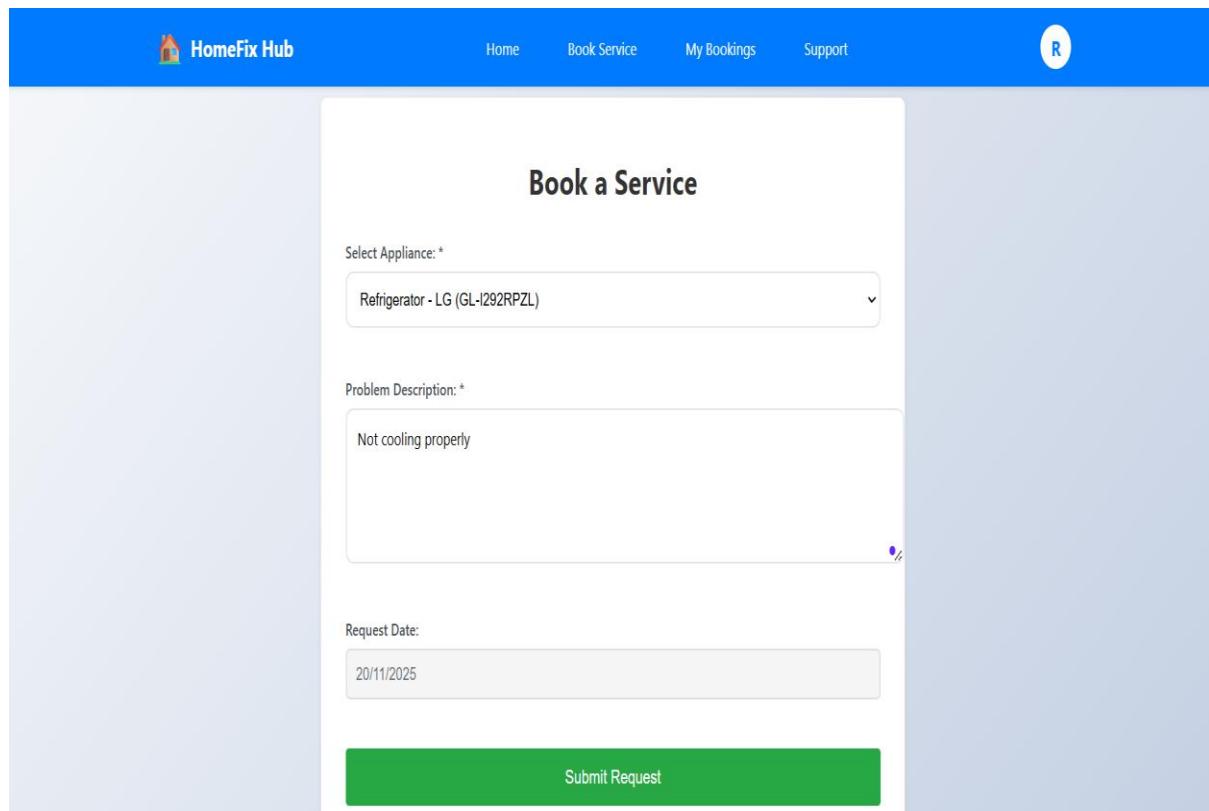


The screenshot shows a "Create Account" form. It contains several input fields with placeholder text and validation messages. The fields include:

- First Name: * (Placeholder: Ashish)
- Last Name: (Placeholder: Kumar B)
- Date of Birth: * (Placeholder: 15-07-2015)
- Address Line 1: * (Placeholder: 555, second floor, 3rd stage, JP nagar)
- Landmark: (Placeholder: Near metro station)
- Stage/Area: (Placeholder: 3rd Street)
- City: * (Placeholder: Bangalore, dropdown menu open)
- Pincode: * (Placeholder: 560085)
- Primary Phone Number: * (Placeholder: 9876543210)
- Secondary Phone Number: (Placeholder: 9120754747)

At the bottom of the form are two buttons: a blue "Create Account" button and a dark gray "Back" button.

3. Book Service:



The screenshot shows the 'Book a Service' page of the HomeFix Hub website. At the top, there's a blue header bar with the 'HomeFix Hub' logo, navigation links for 'Home', 'Book Service', 'My Bookings', and 'Support', and a user profile icon with the letter 'R'. The main content area has a white background with a central form. The title 'Book a Service' is at the top. Below it, a dropdown menu is set to 'Refrigerator - LG (GL-I292RPZL)'. A text input field contains the problem description 'Not cooling properly'. A date input field shows '20/11/2025'. At the bottom is a green 'Submit Request' button.

Book a Service

Select Appliance: *

Refrigerator - LG (GL-I292RPZL)

Problem Description: *

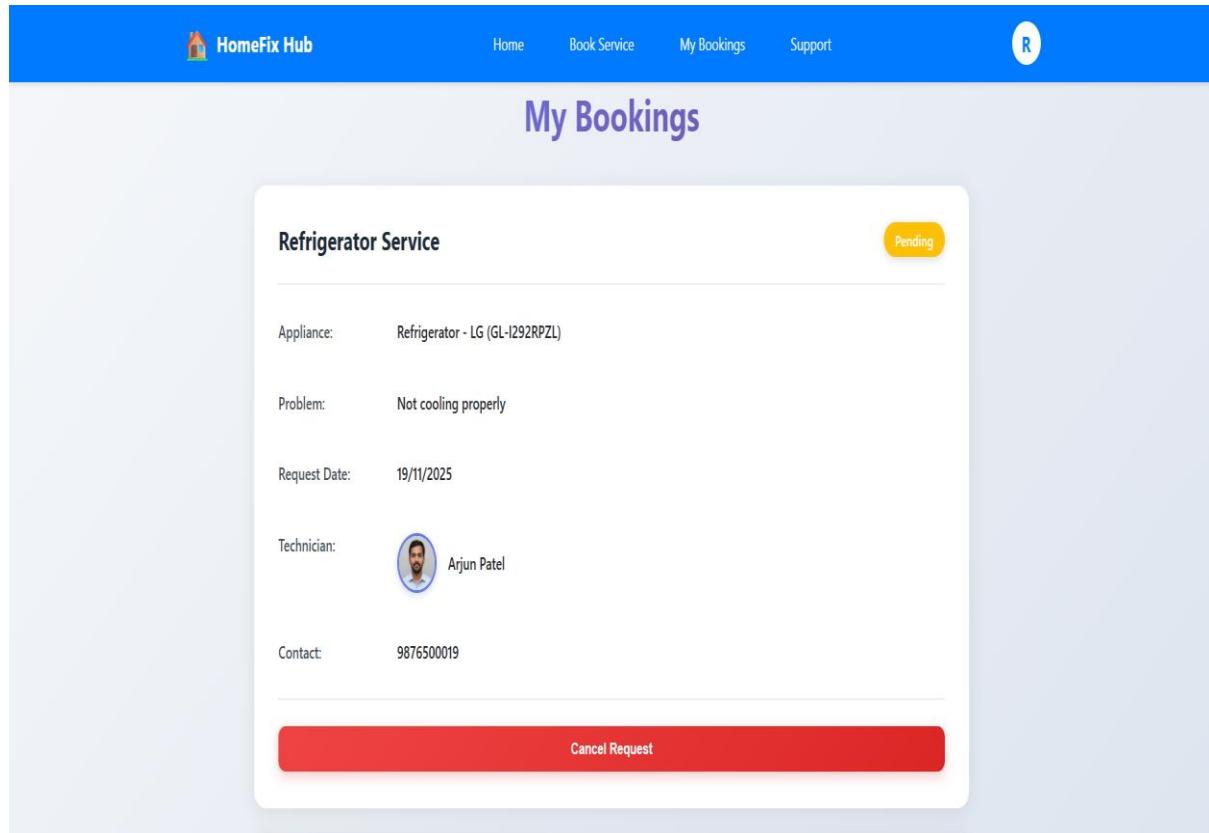
Not cooling properly

Request Date:

20/11/2025

Submit Request

4. My Bookings:



The screenshot shows the 'My Bookings' page of the HomeFix Hub website. The top navigation bar is identical to the previous page. The main content area features a large title 'My Bookings' in purple. Below it, a card displays details for a 'Refrigerator Service'. The card includes fields for 'Appliance' (Refrigerator - LG (GL-I292RPZL)), 'Problem' (Not cooling properly), 'Request Date' (19/11/2025), and 'Technician' (Arjun Patel, with a small profile picture). A red 'Cancel Request' button is at the bottom. A yellow 'Pending' status indicator is located in the top right corner of the service card.

My Bookings

Refrigerator Service

Pending

Appliance: Refrigerator - LG (GL-I292RPZL)

Problem: Not cooling properly

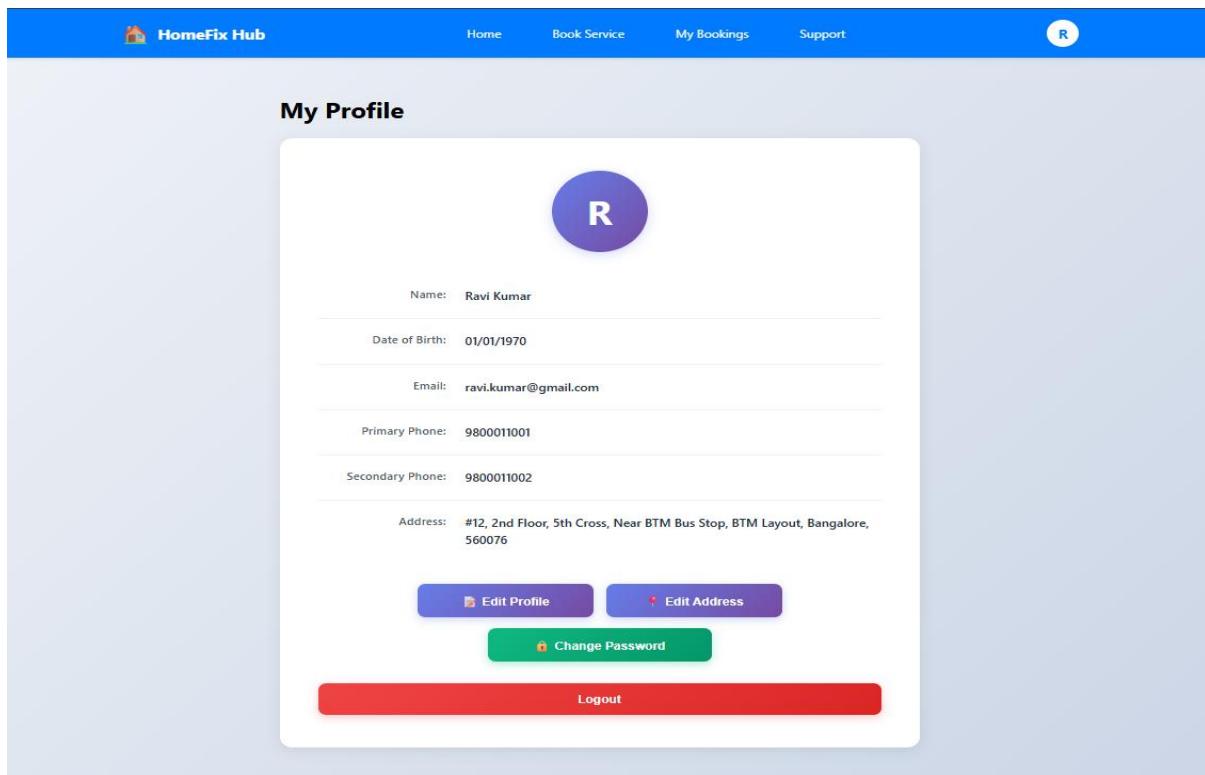
Request Date: 19/11/2025

Technician: Arjun Patel

Contact: 9876500019

Cancel Request

5. Customer Profile:



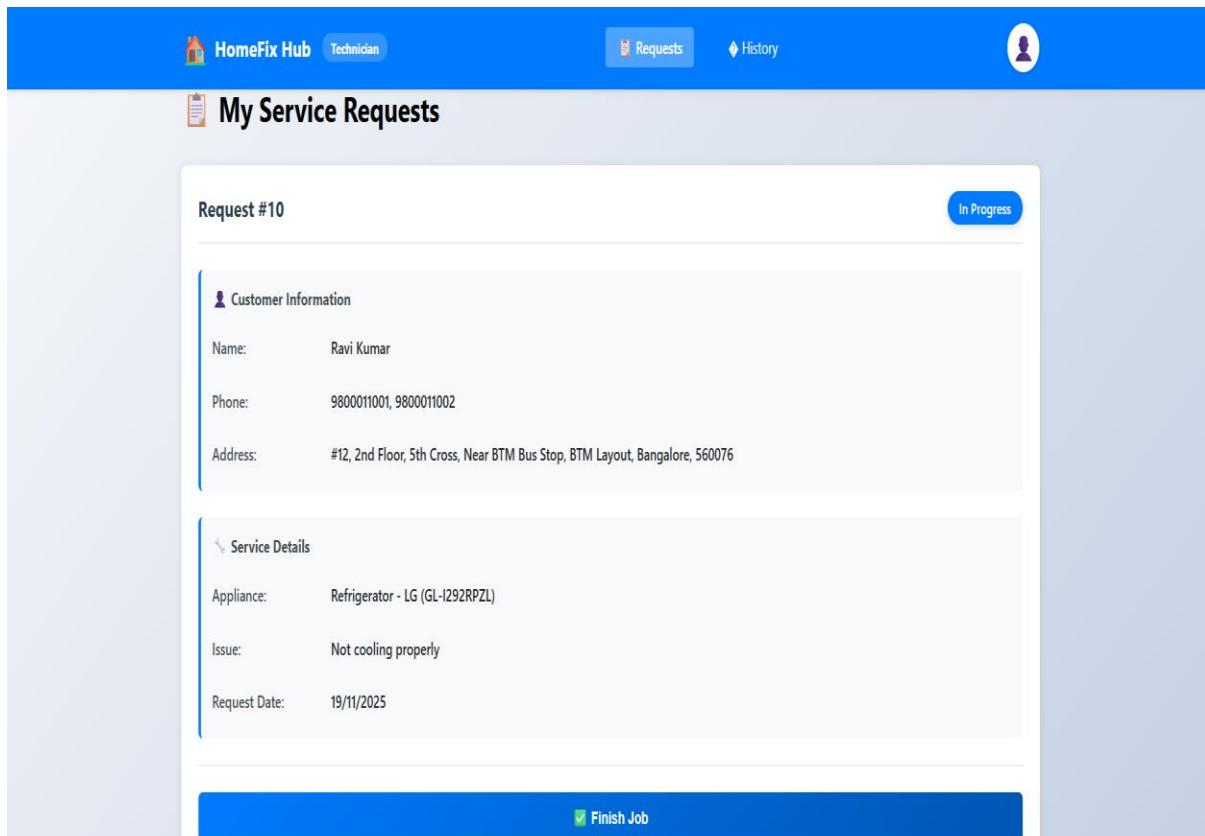
The screenshot shows the 'My Profile' page of the HomeFix Hub application. At the top, there is a blue header bar with the 'HomeFix Hub' logo, navigation links for 'Home', 'Book Service', 'My Bookings', and 'Support', and a user icon with the letter 'R'. Below the header, the page title 'My Profile' is displayed. In the center, there is a large circular placeholder for a profile picture, containing the letter 'R'. Below this, the customer's information is listed in a table format:

| | |
|------------------|---|
| Name: | Ravi Kumar |
| Date of Birth: | 01/01/1970 |
| Email: | ravi.kumar@gmail.com |
| Primary Phone: | 9800011001 |
| Secondary Phone: | 9800011002 |
| Address: | #12, 2nd Floor, 5th Cross, Near BTM Bus Stop, BTM Layout, Bangalore, 560076 |

At the bottom of the profile section, there are three buttons: 'Edit Profile' (purple), 'Edit Address' (purple), and 'Change Password' (green). A red 'Logout' button is located at the very bottom.

Technician portal:

6. Requests:



The screenshot shows the 'My Service Requests' page of the HomeFix Hub application for technicians. At the top, there is a blue header bar with the 'HomeFix Hub' logo, navigation links for 'Technician', 'Requests' (highlighted in blue), 'History' (disabled), and a user icon. Below the header, the page title 'My Service Requests' is displayed. The main content area shows a request details card for 'Request #10':

Request #10 (Status: In Progress)

Customer Information

| | |
|----------|---|
| Name: | Ravi Kumar |
| Phone: | 9800011001, 9800011002 |
| Address: | #12, 2nd Floor, 5th Cross, Near BTM Bus Stop, BTM Layout, Bangalore, 560076 |

Service Details

| | |
|---------------|---------------------------------|
| Appliance: | Refrigerator - LG (GL-I292RPZL) |
| Issue: | Not cooling properly |
| Request Date: | 19/11/2025 |

At the bottom of the card, there is a blue 'Finish Job' button.

7. History:

The screenshot shows the 'Service History' section of the HomeFix Hub app. At the top, there are tabs for 'Requests' and 'History'. Below the tabs, the title 'Service History' is displayed. A card for 'Request #9' is shown, dated 13/11/2025. The card contains sections for 'Customer' (Ravi Kumar, phone 9800011001, address #12, 2nd Floor, 5th Cross, Near BTM Bus Stop, BTM Layout, Bangalore, 560076), 'Service Details' (Appliance: Refrigerator - Whirlpool (ABC123), Issue: Not cooling properly), and 'Invoice Details' (Amount: ₹1500.00, Issue Date: 13/11/2025, Status: Paid). Another card for 'Request #8' is partially visible below it.

8. Technician Profile:

The screenshot shows the technician profile for Arjun Patel. At the top, there are tabs for 'Requests' and 'History'. The profile card features a circular profile picture of Arjun Patel. His name, 'Arjun Patel', and 'Technician ID: #19' are displayed. Key statistics include 'PHONE NUMBER: 9876500019', 'MY RATING: 5.0', 'SERVICE CENTER: FixIt Hub - Bangalore', 'CENTER RATING: 4.53', 'COMPLETED JOBS: 2', and 'ACTIVE JOBS: 1'. Below this, a 'Skills' section lists 'Air Conditioner', 'Refrigerator', and 'Washing Machine' with a '+ Add Skill' button. At the bottom are 'Edit Profile' and 'Logout' buttons.

10. Triggers, Procedures/Functions, Nested query, Join, Aggregate queries:

Triggers, Procedures and Functions is written above under DDL Command section.

Nested Queries:

```
SELECT fname, lname, email
FROM Customer
WHERE customer_id IN (
    SELECT sr.customer_id
    FROM Service_Request sr
    JOIN Invoice i ON sr.request_id = i.request_id
    WHERE i.total_cost > (SELECT AVG(total_cost) FROM Invoice)
);
```

```
SELECT fname, lname, phone_no, rating
```

```
FROM Technician
WHERE center_id = (
    SELECT center_id
    FROM Technician
    GROUP BY center_id
    ORDER BY AVG(rating) DESC
    LIMIT 1
);
```

```
SELECT a.type, a.brand, a.model_no, c.fname
FROM Appliance a
JOIN Customer c ON a.customer_id = c.customer_id
WHERE (a.appliance_id, a.customer_id) NOT IN (
    SELECT appliance_id, customer_id
    FROM Service_Request
);
```

Join Queries:

```
SELECT
    sr.request_id,
    c.fname AS customer_name,
    a.type AS appliance_type,
    a.brand,
    t.fname AS technician_name,
    sr.status,
    sr.request_date
FROM Service_Request sr
INNER JOIN Customer c ON sr.customer_id = c.customer_id
INNER JOIN Appliance a ON sr.appliance_id = a.appliance_id
    AND sr.customer_id = a.customer_id
LEFT JOIN Technician t ON sr.technician_id = t.technician_id;
```

```
SELECT
    t.fname,
    t.lname,
    sc.center_name,
    sc.location,
    GROUP_CONCAT(s.skill) AS skills
FROM Technician t
INNER JOIN Service_Center sc ON t.center_id = sc.center_id
LEFT JOIN Skill s ON t.technician_id = s.technician_id
GROUP BY t.technician_id;
```

```
SELECT
    sr.request_id,
    c.fname AS customer,
    a.type,
    t.fname AS technician,
```

```

    i.total_cost,
    i.payment_status,
    sr.rating
FROM Service_Request sr
INNER JOIN Customer c ON sr.customer_id = c.customer_id
INNER JOIN Appliance a ON sr.appliance_id = a.appliance_id
    AND sr.customer_id = a.customer_id
INNER JOIN Technician t ON sr.technician_id = t.technician_id
INNER JOIN Invoice i ON sr.request_id = i.request_id
WHERE sr.status = 'Completed';

```

Aggregate Queries:

```
SELECT status, COUNT(*) AS total_requests
```

```
FROM Service_Request
```

```
GROUP BY status;
```

```
SELECT
```

```
    sc.center_name,
```

```
    sc.location,
```

```
    COUNT(sr.request_id) AS total_services,
```

```
    SUM(i.total_cost) AS total_revenue,
```

```
    AVG(i.total_cost) AS avg_cost
```

```
FROM Service_Center sc
```

```
INNER JOIN Technician t ON sc.center_id = t.center_id
```

```
INNER JOIN Service_Request sr ON t.technician_id = sr.technician_id
```

```
INNER JOIN Invoice i ON sr.request_id = i.request_id
```

```
GROUP BY sc.center_id;
```

```
SELECT
```

```
    t.fname,
```

```
    t.lname,
```

```
t.rating,  
COUNT(sr.request_id) AS jobs_completed,  
AVG(sr.rating) AS avg_customer_rating,  
SUM(i.total_cost) AS total_earnings  
  
FROM Technician t  
  
LEFT JOIN Service_Request sr ON t.technician_id = sr.technician_id  
AND sr.status = 'Completed'  
  
LEFT JOIN Invoice i ON sr.request_id = i.request_id  
  
GROUP BY t.technician_id  
  
ORDER BY jobs_completed DESC;
```

```
SELECT  
c.fname,  
c.lname,  
COUNT(sr.request_id) AS total_services,  
SUM(i.total_cost) AS total_spent,  
AVG(sr.rating) AS avg_rating_given  
  
FROM Customer c  
  
LEFT JOIN Service_Request sr ON c.customer_id = sr.customer_id  
  
LEFT JOIN Invoice i ON sr.request_id = i.request_id  
  
GROUP BY c.customer_id  
  
HAVING total_services > 0;
```

```
SELECT  
a.type,  
COUNT(sr.request_id) AS service_count  
  
FROM Appliance a  
  
INNER JOIN Service_Request sr ON a.appliance_id = sr.appliance_id  
AND a.customer_id = sr.customer_id  
  
GROUP BY a.type  
  
ORDER BY service_count DESC;
```

11. Code snippets for invoking the Procedures/Functions/Trigger:

Invoking Stored Procedure:

```
CALL get_customer_service_summary(1);
```

Invoking Functions:

```
SELECT get_average_rating(1) AS avg_rating;
```

Invoking Triggers:

```
INSERT INTO Invoice (request_id, total_cost, payment_status, issue_date)
```

```
VALUES (8, 1500.00, 'Unpaid', '2025-11-20');
```

```
SELECT request_id, status FROM Service_Request WHERE request_id = 8;
```

```
UPDATE Service_Request
```

```
SET rating = 4.5
```

```
WHERE request_id = 1;
```

```
SELECT technician_id, rating FROM Technician WHERE technician_id = 1;
```

```
INSERT INTO Service_Request (customer_id, appliance_id, description, request_date, status)
```

```
VALUES (1, 1, 'Refrigerator making noise', '2025-11-20', 'Pending');
```

```
SELECT request_id, customer_id, appliance_id, technician_id, status
```

```
FROM Service_Request
```

```
ORDER BY request_id DESC
```

```
LIMIT 1;
```

```
UPDATE Service_Request
```

```
SET technician_id = NULL
```

```
WHERE request_id = 1 AND status = 'Pending';
```

```
SELECT request_id, technician_id FROM Service_Request WHERE request_id = 1;
```

12. SQL queries(Create, Insert, Triggers, Procedures/Functions, Nested query, Join, Aggregate queries) used in the project in the form of .sql file:

Click [HERE](#) for .sql file

13. Github repo link:

<https://github.com/ashwin921/Home-Appliance-Service-Management-System>