Next Gen. Database System Project

NAME: ABHYUDAYA AGRAWAL

ROLL NO.: 205223001

CLASS: M.Tech 1st year (Data Analytics)

SUBJECT: Next Gen. Database System

PROBLEM STATEMENT

Laundry Management System in NIT-T Campus.

Background:

Laundry facilities are essential services at NIT-T Campus. Managing laundry operations efficiently and providing customers with timely status updates on their laundry orders is a challenge for laundry service providers. To address this, I aim to develop a Laundry Management System (LMS) that streamlines laundry operations, enhances customer experience, and ensures a smooth and efficient laundry processing workflow.

Problem Description:

The Laundry Management System seeks to address the following key problems:

1. Inefficient Laundry Operations: Many laundry facilities still rely on manual processes, leading to inefficiencies in laundry order management, machine scheduling, and resource allocation.

2. Lack of Transparency: Customers often lack real-time information about the status of their laundry orders, leading to frustration and inconvenience.

3. Order Tracking Challenges: Tracking the status of laundry orders, from drop-off to pick-up, can be challenging for both customers and service providers.

Solution Objectives:

The Laundry Management System aims to achieve the following objectives:

1. Efficient Operations: Automate laundry order management, machine scheduling, and resource allocation to optimize laundry processing.

2. Customer Transparency: Provide customers with real-time updates on the status of their laundry orders through a user-friendly interface.

3. Order Tracking: Enable easy tracking of laundry orders at every stage, from order placement to delivery or pick-up.

Key Features:

1. **Place Order:**
    - Users can easily place laundry orders through the system.
    - The system provides a user-friendly interface to select types of clothes, specify quantities, and choose service preferences.
2. **User Authentication:**
    - Secure user authentication ensures that only authorized users can access the system.
    - Users can register, log in, and manage their profiles.
3. **Order Tracking:**
    - Customers can track the status of their laundry orders using a unique reference ID.
    - Real-time updates on the order status provide transparency and convenience.
4. **Admin Dashboard:**
    - An admin dashboard allows staff to manage and monitor all incoming orders.
    - Admins can update order statuses, view customer details, and handle order-related tasks.
5. **Contact Us:**
    - The system includes a contact form for users to reach out with inquiries or support requests.

Technologies Used:

- **Frontend:** HTML, Bootstrap
- **Backend:** Flask (Python)
- **Database:** MongoDB for user registration and order management
- **Authentication:** Password hashing for user security
- **Styling:** Bootstrap for a responsive and visually appealing design

How to Use:

1. **Place Order:**
    - Visit the "Place Order" page to submit a laundry order, specifying the types of clothes and service preferences.
2. **User Authentication:**
    - New users can register by providing necessary details.
    - Returning users can log in to access their account.
3. **Order Tracking:**
    - Use the "Check Status" page with the reference ID to track the status of a laundry order.
4. **Admin Dashboard:**

- Admins can log in to the dashboard to manage orders, update statuses, and view customer details.

5. **Contact Us:**
    - Reach out through the "Contact Us" page for any assistance or inquiries.

Benefits:

- **Efficiency:** Streamlines the laundry ordering process for both users and service providers.
- **Transparency:** Real-time order tracking enhances customer satisfaction.
- **User-Friendly:** Intuitive interfaces make it easy for users to navigate and interact with the system.
- **Security:** Password hashing ensures the security of user accounts and information.

List of frameworks, libraries, and dependencies along with their installation instructions:

Backend (Flask - Python Framework): Flask: Installation: pip install Flask

Flask-PyMongo (for MongoDB integration): Installation: pip install Flask-PyMongo

Passlib (for password hashing): Installation: pip install passlib

Database: MongoDB: Follow the official instructions for installing MongoDB.

Note:

Make sure to replace placeholders like 'your_secret_key', 'your-mongodb-url', 'username', and 'password' with your actual values.

COMPLETE DATABASE SCHEMA

1). Table: User Model

Column Name	Data Type	Constraints
User_id	INT	PRIMARY KEY, Auto Increment
Username	VARCHAR(150)	UNIQUE, NOT NULL
Password	VARCHAR(128)	NOT NULL
FirstName	VARCHAR(250)	NOT NULL
LastName	VARCHAR(255)	NOT NULL
EMAIL	VARCHAR(40)	UNIQUE, NOT NULL
Phone	VARCHAR(10)	NOT NULL
Address	VARCHAR(255)	NOT NULL
AccountBalance	DECIMAL(10,2)	DEFAULT 0.00

2). Table: Orders Model

Column Name	Data Type	Constraints
Order_id	INT	PRIMARY KEY, Auto Increment
User_id	INT	NOT NULL, FOREIGN KEY
OrderDate	DATE	NOT NULL
DueDate	DATE	NOT NULL
TotalAmount	DECIMAL(10,2)	NOT NULL
OrderStatus	VARCHAR(20)	NOT NULL (eg. 'PENDING', 'COMPLETED')

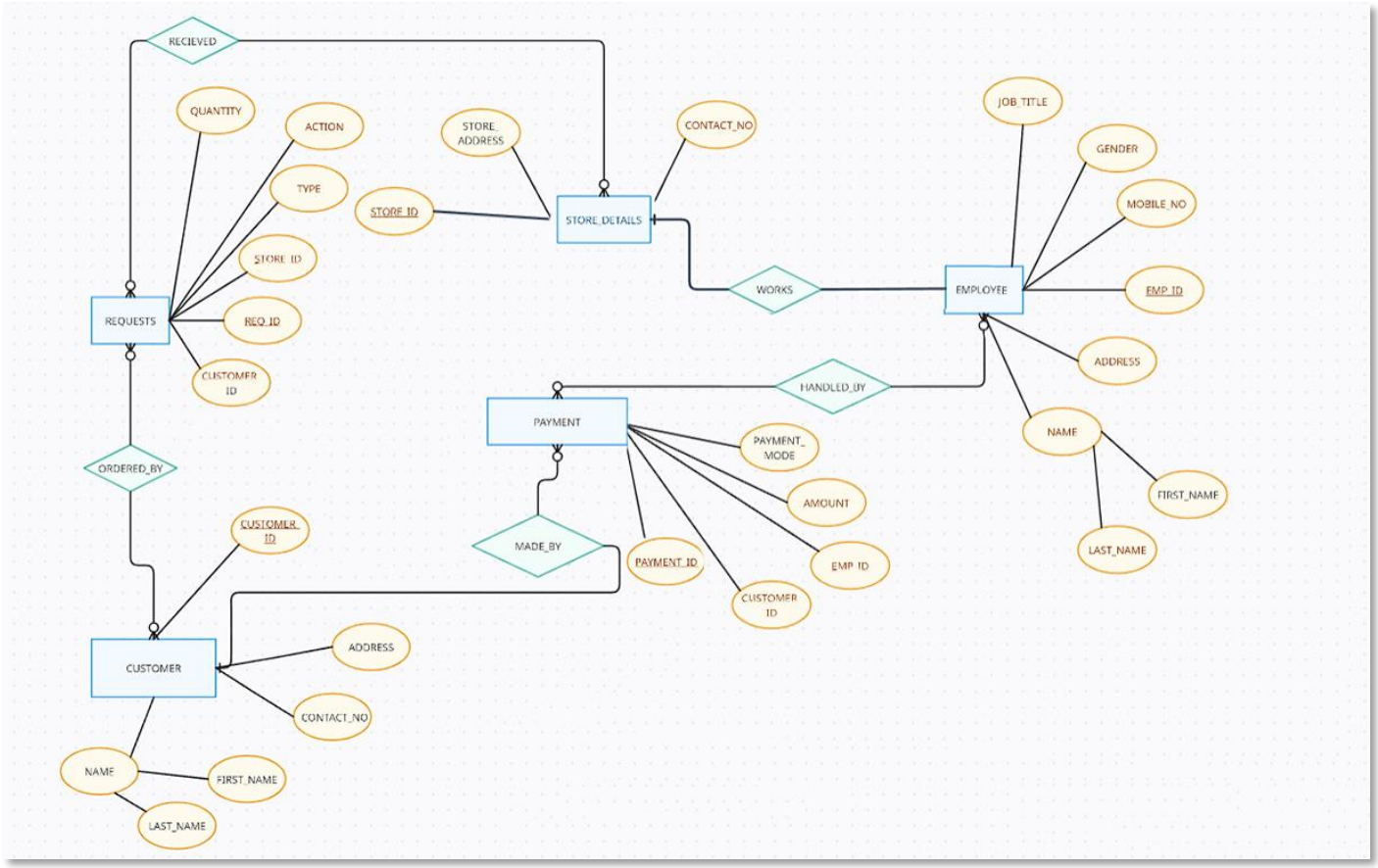
3). Table: Laundry Order Items Model

Column Name	Data Type	Constraints
OrderItemID	INT	PRIMARY KEY, AUTO INCREMENT
Order_id	INT	NOT NULL, FOREIGN KEY
LaundryItemID	INT	NOT NULL, FOREIGN KEY
Quantity	INT	NOT NULL, CHECK (QUANTITY >=1)
PricePerUnit	DECIMAL(10,2)	NOT NULL, CHECK (PricePerUnit >=0)
TotalPrice	DECIMAL(10,2)	NOT NULL, CHECK (TotalPrice >=0)

4). Table: Payment Model

Column Name	Data Type	Constraints
PaymentID	INT	PRIMARY KEY, AUTO INCREMENT
Order_id	INT	NOT NULL, FOREIGN KEY
PaymentDate	DATETIME	NOT NULL
Amount	INT	NOT NULL
PaymentMethod	VARCHAR(20)	NOT NULL (eg, 'CASH')
PaymentStatus	VARCHAR(20)	NOT NULL (eg. 'PAID', 'PENDING')

ENTITY RELATIONSHIP DIAGRAM



LIST OF STAKEHOLDERS

1. **Admin/Owner:**
    - Individuals or entities that oversee the operation of the laundry facility.
2. **Employee:**
    - Staff
3. **Customer:**
    - Students (End Users)

DATABASE SCHEMA FOR EACH STAKE-HOLDER

Admins:

Table Name	Fields	Description
Admin	**id** (Primary Key), **username**, **email**, **password, name**...	Information about administrators.
Order	**Order_id** (Primary Key), **Customer_id** (Foreign Key),**Amount** , **Order_date**, ...	Contains Details of laundry order
Report	**id** (Primary Key), **admin_id** (Foreign Key), **report_text**, **date_generated**, ...	Reports and analytics generated by administrators.

Employee:

Table Name	Fields	Description
Staff	**EmpId** (Primary Key), **username**, **email**, **password**, ...	Information about Staff.
LaundryOrder	**Order_id** (Primary Key), **Customer_id** (Foreign Key),**Amount** , **Order_date**, ...	Contains details of laundry orders

Customer:

Table Name	Fields	Description
Customer	**id** (Primary Key), **username**, **email**, **password**, ...	Stores customer information (e.g., name, contact)
LaundryRequest	**RequestId** (Primary Key), **Customer_id** (Foreign Key), **RequestDate**, **Satus**, ...	Records laundry service requests by customers
PaymentTransaction	**PaymentId** (Primary Key), **Customer_Id** (Foreign Key), **TransationDate**, **Amount**, **PaymentMethod**, ...	Tracks payment transactions by customers

DATABASE USED AND THEIR VIEWS

DATABASE: Mongo DB and MySQL

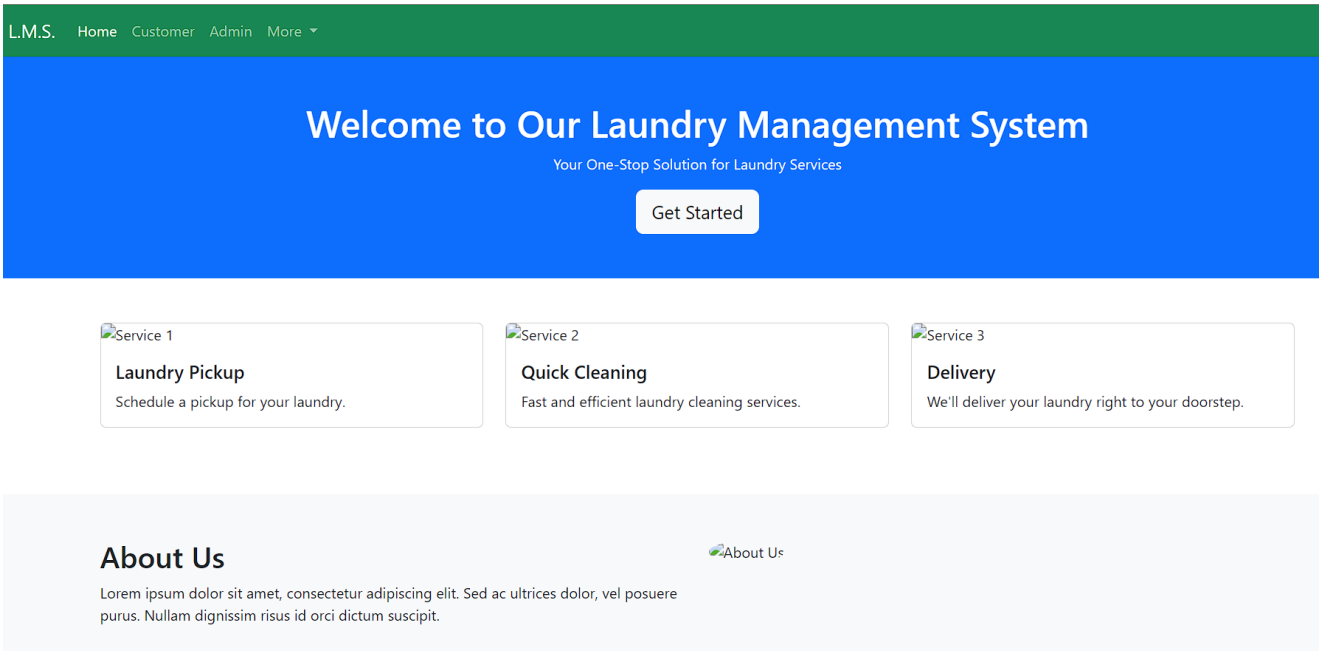
DATABASE VIEWS:

1. **Admin:**
    - **Admins Collection:**
        - Fields:
            - **admin_id** (unique identifier)
            - **username** (admin_username)
            - **email** (admin@example.com)
            - **password** (hashed password)
            - **first_name** (name)
            - **phone** (phone)
            - Other admin-specific information
2. **Staff:**
    - **Employees Collection:**
        - Fields:
            - **emp_id** (unique identifier)
            - **username** (employee’s username)
            - **email** (employee's email)
            - **password** (hashed password)
            - Other employee-specific information
3. **Customer:**
    - **Customer Collection:**
        - Fields:
            - **customer_id** (unique identifier)
            - **username** (customer's username)
            - **email** (customer's email)

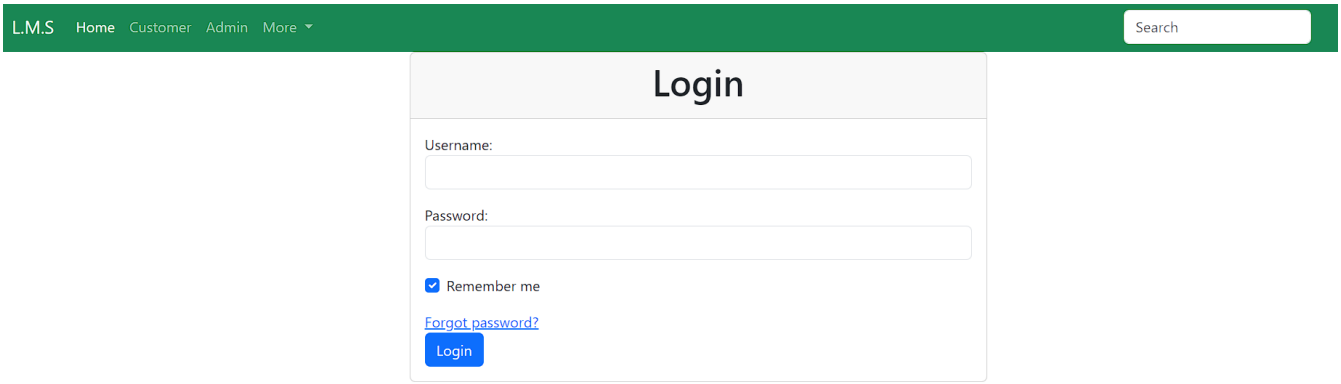
- **password** (hashed password)
- Other student-specific information

FRONTEND SCREENSHOTS

1). index.html - home

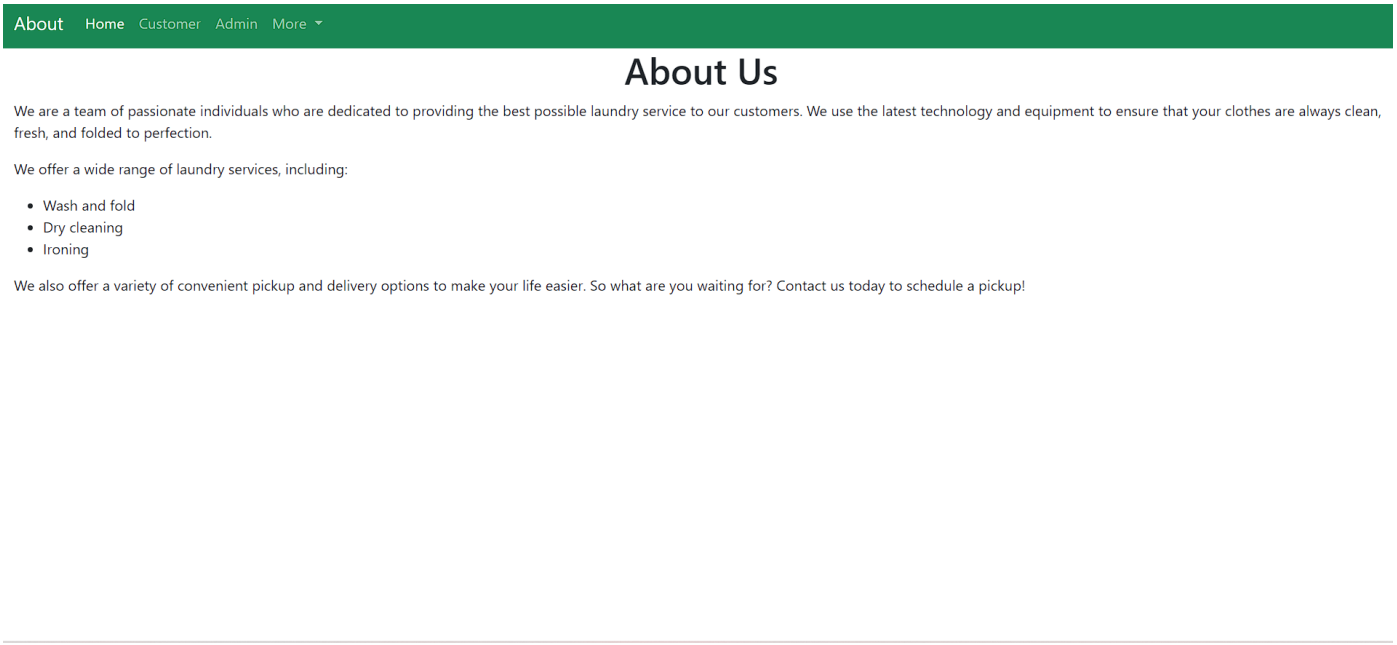


2). login.html - login page

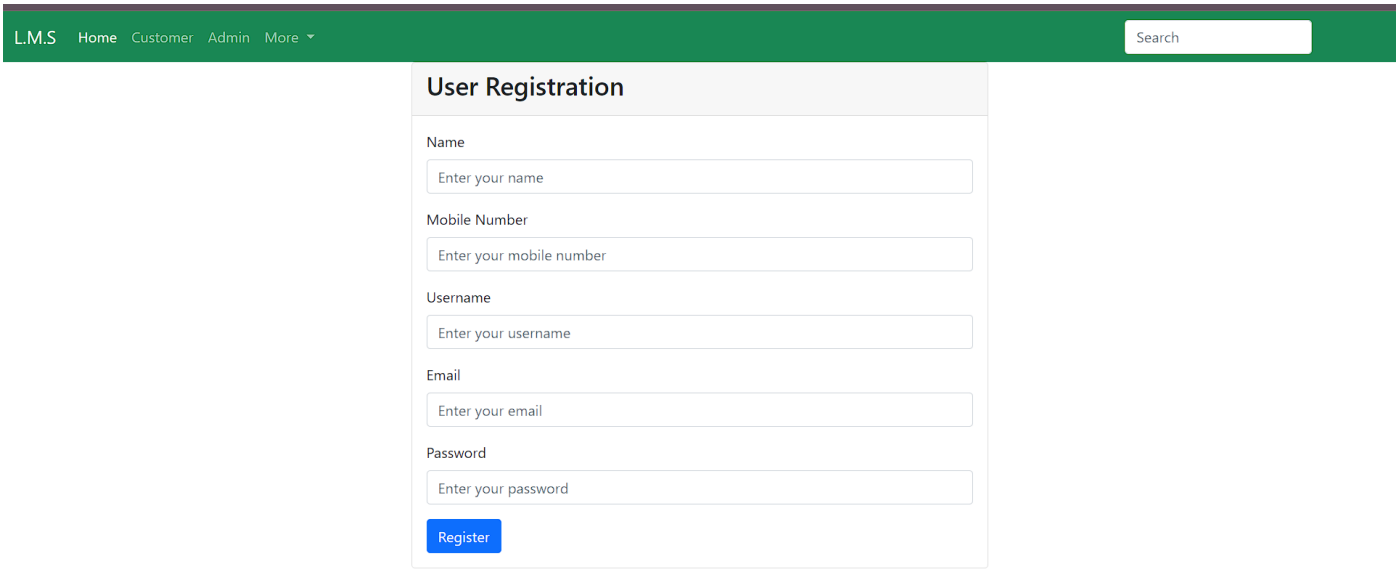


3). admin.html – Admin Dashboard

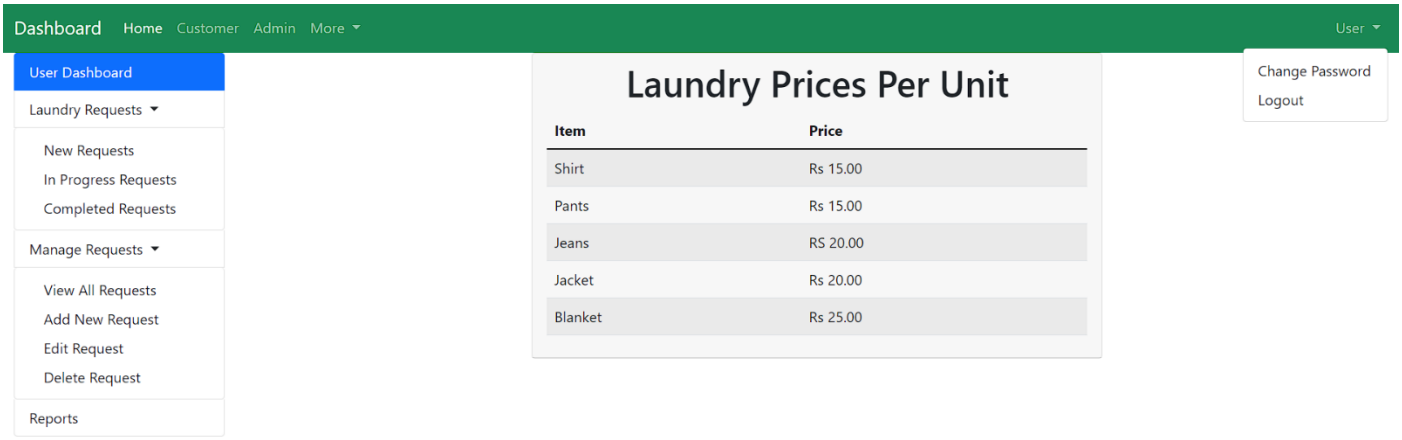
4). about.html -> About page



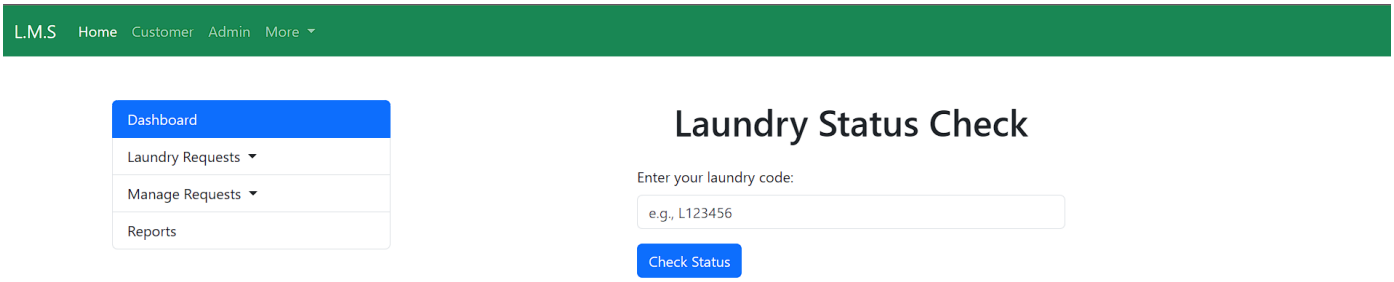
5). reg.html -> User Registration page



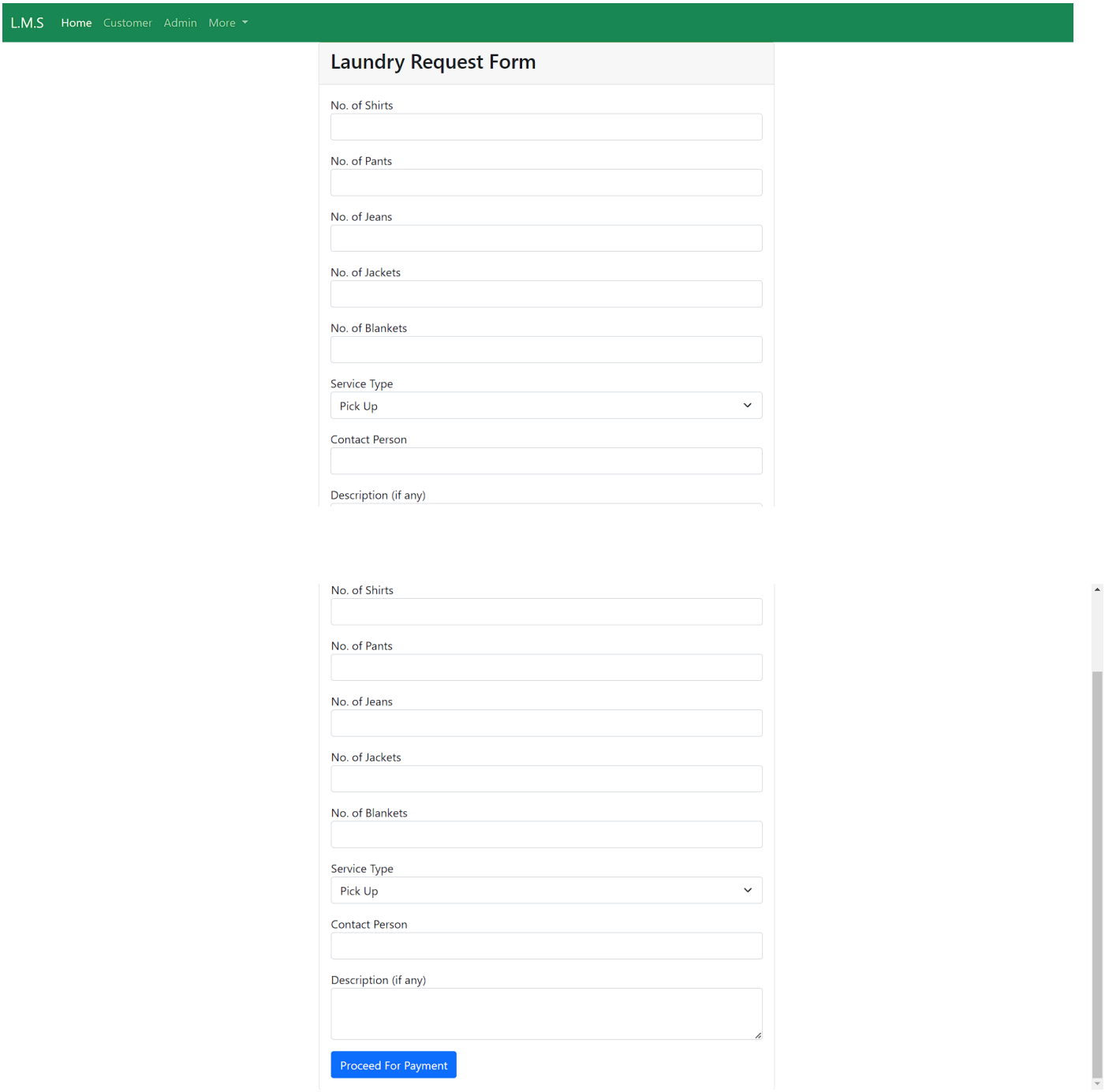
6). User.html -> User Dashboard page



7). status.html -> Status Check page



8). Request.html -> Laundry Request page


L.M.S
Home
Customer
Admin
More
Laundry Request Form
No. of Shirts
No. of Pants
No. of Jeans
No. of Jackets
No. of Blankets
Service Type
Pick Up
Contact Person
Description (if any)

No. of Shirts
No. of Pants
No. of Jeans
No. of Jackets
No. of Blankets
Service Type
Pick Up
Contact Person
Description (if any)
Proceed For Payment