

# EQUIPMENT BOOKING SYSTEM

**A Mini Project Report**

**Submitted by DHANUSHRI M 23CDR033 DIVYA C S 23CDR043**

**22CDT41 - DATABASE MANAGEMENT SYSTEMS**

**DEPARTMENT OF COMPUTER SCIENCE AND DESIGN**

# KONGU ENGINEERING COLLEGE

## (Autonomous)

**PERUNDURAI ERODE-638060**

# MAY 2025

**Problem Statement:**

Allows users to reserve shared equipment within an organization. (RDBMS: Equipment, Reservations, Users; Backend: Equipment availability, booking management).

# Github Link:

<https://github.com/dhanushridesign/equipment> <https://github.com/divyawebdesign/equipmentbooking>

# Modules and Description:

1. **Admin Signup/Login Module**

This module allows the admin to create an account and log in securely. Only authenticated admins can access the dashboard to manage equipment data and view bookings.

1. **User Signup/Login Module**

Users can register with their details and log in to the system. The module validates credentials and provides access to available equipment for booking.

1. **Admin Dashboard Module**

After logging in, the admin can view all equipment records. From here, they can add new equipment, edit existing entries, or delete unused equipment.

1. **Equipment Management Module**

This module stores and manages equipment data including name, description, quantity, category, and current status (available/booked).

1. **Booking Module**

Users can select equipment and submit a booking request. The system stores the booking details and links it with the user and equipment IDs.

1. **Category Filter Module (User Side)**

Users can filter the available equipment based on categories like electronics, sports, etc., making it easier to find relevant items.

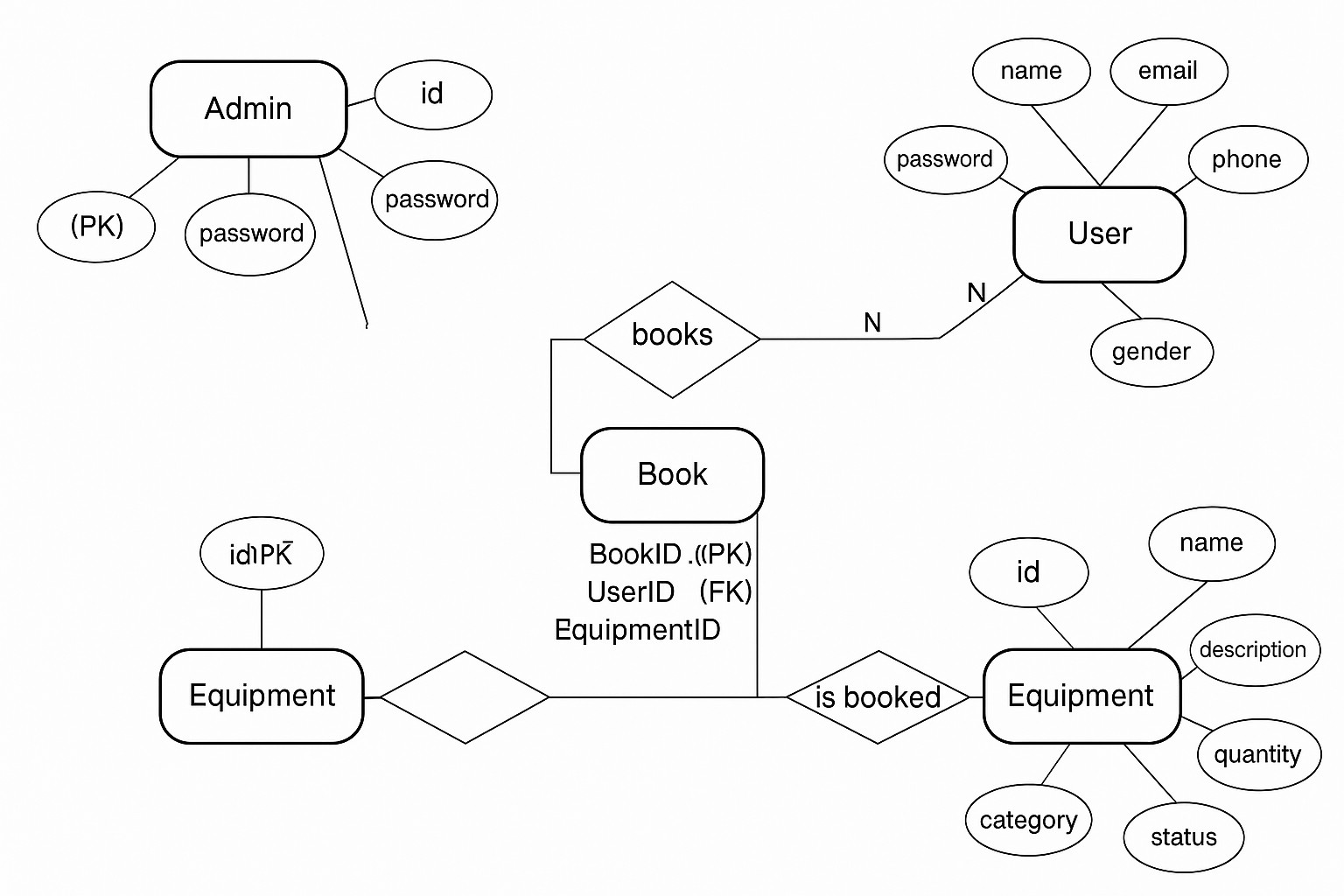
1. **Delete and Edit Equipment Module (Admin Side)**

Admins can update equipment information or remove entries that are no longer needed, helping to keep the inventory up to date.

1. **Alerts and Validations**

This module provides frontend alerts like "signup successful" or "passwords don’t match," and ensures all required fields are properly validated before submission.

# ER Diagram:



**Programming Tools:**

## Frontend (User Interface)

These tools are used to create and design the visible part of the website:

## HTML (HyperText Markup Language):

Used to create the structure of web pages like login forms, signup pages, dashboards, etc.

## CSS (Cascading Style Sheets):

Used to style the HTML content — for example, to add colors, spacing, fonts, and layout.

## JavaScript (Basic):

Used to add interactive behavior like alert messages and input validation in forms.

## Jinja2 (Template Engine):

Built into Flask, used to dynamically insert data into HTML pages from the backend (like showing a list of equipment).

## Backend (Server-side Logic)

These tools power the functionality behind the scenes (like logging in, storing data, processing requests):

## Python:

The main programming language used for writing backend logic.

## Flask (Web Framework):

A lightweight Python web framework used to handle URL routing, form submission, connecting to the database, and rendering HTML templates.

## Flask-MySQLdb (Connector):

A library that allows the Flask app to interact with a MySQL database (insert, fetch, delete data).

## Database (Data Storage)

These tools store and manage data like user details, equipment information, and bookings:

## MySQL:

A popular relational database management system used to store data in tables like users, admin, equipment, and book.

## Visual Studio Code (VS Code)

A powerful and user-friendly code editor used to write and manage the project’s HTML, CSS, Python, and JavaScript files. It also supports syntax highlighting, extensions, and integrated terminal for running the Flask server.

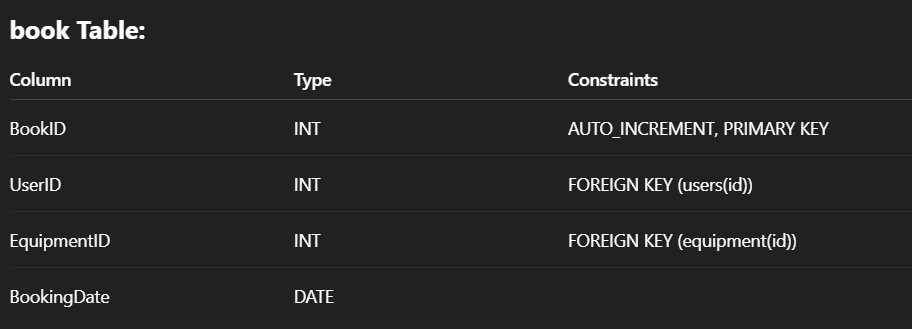
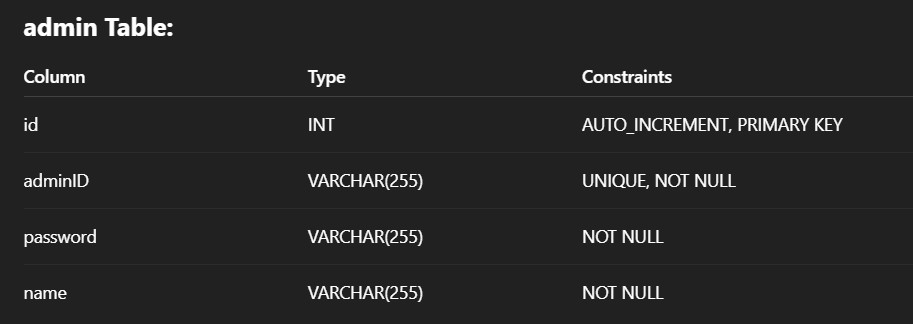
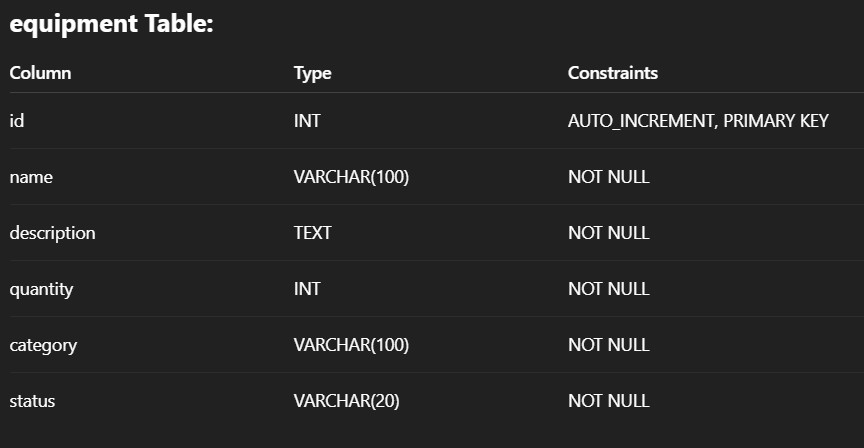
## Terminal / Command Prompt

Used to run the Flask application, install Python packages, and start the development server with commands like python [app.py](http://app.py/).

## Google Chrome (or any modern browser)

Used to open and test the web pages of the Equipment Booking System, such as login forms, dashboards, and equipment listings.

# Database Design:



**SQL Query:**

* + - Equipment table

CREATE TABLE equipment (id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(100) NOT NULL, description TEXT NOT NULL, quantity INT NOT NULL, category VARCHAR(100) NOT NULL, status VARCHAR(20) NOT NULL);

* + - User table

CREATE TABLE users (id INT AUTO\_INCREMENT PRIMARY KEY, name VARCHAR(100), email VARCHAR(100) UNIQUE, password

VARCHAR(255), phone VARCHAR(20), gender ENUM('female', 'male', 'other'), created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP);

* + - Admin Table

CREATE TABLE admin (

id INT NOT NULL AUTO\_INCREMENT, adminID VARCHAR(255) NOT NULL UNIQUE, password VARCHAR(255) NOT NULL,

name VARCHAR(255) NOT NULL, PRIMARY KEY (id)

);

* + - Book Table

CREATE TABLE book (

BookID INT NOT NULL AUTO\_INCREMENT, UserID INT,

EquipmentID INT, BookingDate DATE, PRIMARY KEY (BookID),

FOREIGN KEY (UserID) REFERENCES user(UserID),

FOREIGN KEY (EquipmentID) REFERENCES equipment(EquipmentID)

);

# Coding:

[**App.py**](http://app.py/)

from flask import Flask, render\_template, request, redirect, url\_for from flask\_mysqldb import MySQL

app = Flask(\_name\_) app.config['MYSQL\_HOST'] = 'localhost' app.config['MYSQL\_USER'] = 'divya'

app.config['MYSQL\_PASSWORD'] = 'Strong!password' app.config['MYSQL\_DB'] = 'equipment'

mysql = MySQL(app) @app.route('/')

def home():

return redirect(url\_for('render\_login\_page')) @app.route('/admin\_signup', methods=['GET', 'POST']) def admin\_signup():

if request.method == 'GET':

return render\_template('admin\_signup.html') else:

admin\_id = request.form['adminID'] password = request.form['password']

confirm\_password = request.form['confirmPassword'] name = request.form['name']

if not admin\_id or not password or not name:

return "All fields are required!", 400 if password != confirm\_password:

return "Passwords do not match!", 400 try:

cur = mysql.connection.cursor()

cur.execute("INSERT INTO admin (adminID, password, name) VALUES (%s, %s, %s)", (admin\_id, password, name))

mysql.connection.commit() cur.close()

return "Signup successful! <a href='/admin\_login.html'>Login now</a>" except Exception as e:

return f"Database error: {str(e)}", 500 @app.route('/admin\_login', methods=['POST']) def admin\_login():

admin\_id = request.form['adminID'] password = request.form['password'] cur = mysql.connection.cursor()

cur.execute("SELECT \* FROM admin WHERE adminID = %s AND password = %s", (admin\_id, password))

admin = cur.fetchone() cur.close()

if admin:

return redirect(url\_for('admin\_dashboard')) else:

return "Login failed. <a href='/admin\_signup'>Sign up here</a>" @app.route('/admin\_login.html')

def render\_login\_page():

return render\_template('admin\_login.html') @app.route('/admin\_dashboard')

def admin\_dashboard():

cur = mysql.connection.cursor() cur.execute("SELECT \* FROM equipment") equipment\_list = cur.fetchall()

cur.close()

return render\_template('admin\_dashboard.html', equipment\_list=equipment\_list, admin\_name="Admin")

@app.route('/add\_equipment', methods=['GET', 'POST']) def add\_equipment():

if request.method == 'GET':

return render\_template('add\_equipment.html') else:

equipment\_name = request.form['equipmentName'] description = request.form['description']

quantity = request.form['quantity'] category = request.form['category'] status = request.form['status']

try:

cur = mysql.connection.cursor()

cur.execute("INSERT INTO equipment (name, description, quantity, category, status) VALUES (%s, %s, %s, %s, %s)", (equipment\_name, description, quantity, category, status))

mysql.connection.commit() cur.close()

return redirect(url\_for('admin\_dashboard')) except Exception as e:

return f"Database error: {str(e)}", 500 @app.route('/delete\_equipment', methods=['GET', 'POST']) def delete\_equipment():

cur = mysql.connection.cursor() cur.execute("SELECT \* FROM equipment") equipment\_list = cur.fetchall()

cur.close()

if request.method == 'POST':

equipment\_id = request.form['equipment\_id'] try:

cur = mysql.connection.cursor()

cur.execute("DELETE FROM equipment WHERE id = %s", (equipment\_id,)) mysql.connection.commit()

cur.close()

return redirect(url\_for('delete\_equipment')) except Exception as e:

return f"Database error: {str(e)}", 500

return render\_template('delete\_equipment.html', equipment\_list=equipment\_list)

@app.route('/update\_equipment') def update\_equipment():

cur = mysql.connection.cursor() cur.execute("SELECT \* FROM equipment") equipment\_list = cur.fetchall()

cur.close()

return render\_template('update-equipment.html', equipment\_list=equipment\_list) @app.route('/edit\_equipment/<int:equipment\_id>', methods=['GET', 'POST'])

def edit\_equipment(equipment\_id):

cur = mysql.connection.cursor()

cur.execute("SELECT \* FROM equipment WHERE id = %s", (equipment\_id,)) equipment = cur.fetchone()

cur.close()

if request.method == 'POST':

equipment\_name = request.form['equipmentName'] description = request.form['description']

quantity = request.form['quantity'] category = request.form['category'] status = request.form['status']

try:

cur = mysql.connection.cursor()

cur.execute("UPDATE equipment SET name = %s, description = %s, quantity = %s, category = %s, status = %s WHERE id = %s", (equipment\_name, description, quantity, category, status, equipment\_id))

mysql.connection.commit() cur.close()

return redirect(url\_for('admin\_dashboard')) except Exception as e:

return f"Error: {str(e)}", 500

return render\_template('edit-equipment.html', equipment=equipment) @app.route('/user\_signup', methods=['GET', 'POST'])

def user\_signup():

if request.method == 'GET':

return render\_template('user\_signup.html') else:

name = request.form['name'] email = request.form['email']

password = request.form['password'] confirm\_password = request.form['confirmPassword'] phone = request.form['phone']

gender = request.form['gender']

if not name or not email or not password or not phone or not gender: return "All fields are required!", 400

if password != confirm\_password:

return "Passwords do not match!", 400 try:

cur = mysql.connection.cursor()

cur.execute("INSERT INTO users (name, email, password, phone, gender) VALUES (%s, %s, %s,

%s, %s)", (name, email, password, phone, gender)) mysql.connection.commit()

cur.close() return '''

<script>

alert("Signup successful!"); window.location.href = "/user\_login.html";

</script>

'''

except Exception as e:

return f"Database error: {str(e)}", 500 @app.route('/user\_login', methods=['POST']) def user\_login():

email = request.form['email'] password = request.form['password'] cur = mysql.connection.cursor()

cur.execute("SELECT \* FROM users WHERE email = %s AND password = %s", (email, password)) user = cur.fetchone()

cur.close() if user:

return redirect(url\_for('user\_dashboard', username=user[1], user\_id=user[0])) else:

return "Login failed. <a href='/user\_signup'>Sign up here</a>" @app.route('/user\_login.html')

def render\_user\_login\_page():

return render\_template('user\_login.html') @app.route('/user\_dashboard/<username>') def user\_dashboard(username):

category = request.args.get('category', 'all') cur = mysql.connection.cursor()

if category == 'all':

cur.execute("SELECT \* FROM equipment") else:

cur.execute("SELECT \* FROM equipment WHERE category = %s", (category,)) equipment\_list = cur.fetchall()

cur.close()

return render\_template('user-dashboard.html', username=username, equipment\_list=equipment\_list, selected\_category=category) @app.route('/book\_equipment', methods=['POST'])

def book\_equipment():

user\_id = request.form['user\_id'] equipment\_id = request.form['equipment\_id'] try:

cur = mysql.connection.cursor()

cur.execute("INSERT INTO book (UserID, EquipmentID) VALUES (%s, %s)", (user\_id, equipment\_id)) mysql.connection.commit()

cur.close() return '''

<script>

alert("Booking request submitted successfully!");

</script>

'''

except Exception as e:

return f"Database error: {str(e)}", 500 if \_name\_ == '\_main\_':

app.run(debug=True)

add\_equipment.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0"/>

<title>Add Equipment</title>

<style> body {

font-family: 'Gill Sans', Calibri, 'Trebuchet MS', sans-serif; background-color: #E2DFD0;

margin: 0;

padding: 0; display: flex;

align-items: center; justify-content: center; height: 100vh;

}

.form-container { background-color: #fff; padding: 40px;

border-radius: 12px;

box-shadow: 0 6px 18px rgba(0, 0, 0, 0.2); max-width: 450px;

width: 100%;

}

h2 {

text-align: center; color: #32012F; margin-bottom: 30px;

}

label {

display: block; margin-bottom: 8px; color: #524C42;

font-weight: bold;

}

input[type="text"], input[type="number"], select {

width: 100%; padding: 10px; margin-bottom: 20px; border-radius: 8px;

border: 1px solid #ccc; font-size: 16px;

}

button { width: 100%;

padding: 12px;

background-color: #32012F; color: #E2DFD0;

border: none; border-radius: 8px; font-size: 18px; cursor: pointer;

transition: background-color 0.3s ease;

}

button:hover {

background-color: #F97300;

}

.back-link { display: block; text-align: center; margin-top: 20px; color: #32012F;

text-decoration: none; font-weight: bold;

}

.back-link:hover {

text-decoration: underline;

}

</style>

</head>

<body>

<div class="form-container">

<h2>Add Equipment</h2>

<form action="/add\_equipment" method="POST">

<label for="equipmentName">Equipment Name:</label>

<input type="text" id="equipmentName" name="equipmentName" required>

<label for="description">Description:</label>

<input type="text" id="description" name="description" required>

<label for="quantity">Quantity:</label>

<input type="number" id="quantity" name="quantity" required min="1">

<label for="category">Category:</label>

<select id="category" name="category" required>

<option value="agriculture">Agriculture</option>

<option value="construction">Construction</option>

<option value="industrial">Industrial</option>

<option value="transportation">Transportation</option>

</select>

<label for="status">Status:</label>

<select id="status" name="status" required>

<option value="Available">Available</option>

<option value="Unavailable">Unavailable</option>

</select>

<button type="submit">Add Equipment</button>

</form>

<a class="back-link" href="{{ url\_for('admin\_dashboard') }}">← Back to Dashboard</a>

</div>

</body>

</html>

admin\_dashboard.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0"/>

<title>Admin Dashboard</title>

<style> body { margin: 0;

font-family: 'Gill Sans', Calibri, 'Trebuchet MS', sans-serif; background-color: #E2DFD0;

color: #524C42;

}

header {

background-color: #32012F; padding: 25px;

text-align: center; color: #E2DFD0; font-size: 32px; font-weight: bold;

box-shadow: 0 4px 10px rgba(0, 0, 0, 0.2);

}

.dashboard {

max-width: 1200px; margin: 40px auto; padding: 20px;

}

.stats-grid { display: grid;

grid-template-columns: repeat(auto-fit, minmax(220px, 1fr)); gap: 25px;

margin-bottom: 50px;

}

.stat-card {

background-color: #fff; padding: 25px;

border-radius: 12px;

box-shadow: 0 6px 16px rgba(0, 0, 0, 0.1); text-align: center;

}

.stat-card h3 { font-size: 22px;

margin-bottom: 10px; color: #32012F;

}

.stat-card p { font-size: 28px;

font-weight: bold; color: #F97300;

}

.button-grid { display: grid;

grid-template-columns: repeat(3, 1fr); gap: 20px;

justify-items: center; margin-bottom: 50px;

}

.dashboard-button { background-color: #32012F; color: #E2DFD0;

padding: 20px; border: none; border-radius: 12px; font-size: 18px; cursor: pointer;

transition: background-color 0.3s;

width: 100%;

max-width: 220px; text-align: center;

}

.dashboard-button:hover { background-color: #F97300;

}

.about-section { background-color: #fff; padding: 40px;

border-radius: 14px;

box-shadow: 0 4px 12px rgba(0, 0, 0, 0.1);

}

.about-section h2 { color: #32012F; margin-bottom: 20px;

}

.about-section p { font-size: 18px; line-height: 1.6;

}

</style>

</head>

<body>

<header>

Welcome, {{ admin\_name }}! | Admin Dashboard

</header>

<div class="dashboard">

<!-- Action Buttons -->

<div class="button-grid">

<button class="dashboard-button" onclick="location.href='/add\_equipment'">Add Equipment</button>

<button class="dashboard-button" onclick="location.href='/delete\_equipment'">Delete Equipment</button>

<button class="dashboard-button" onclick="location.href='/view\_bookings'">View Bookings</button>

<button class="dashboard-button" onclick="location.href='/update\_equipment'">Update Equipment</button>

<button class="dashboard-button" onclick="location.href='/manage\_users'">Manage Users</button>

</div>

<!-- About Section -->

<div class="about-section">

<h2>About the Equipment Booking System</h2>

<p>

This dashboard is designed to help administrators manage equipment reservations within an organization.

You can quickly add, delete, view, or update equipment records, monitor bookings, and manage user access.

The system ensures that resources are used efficiently, reduces manual paperwork, and gives you full control

over organizational assets.

</p>

</div>

</div>

</body>

</html>

admin\_login.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0"/>

<title>Admin Login</title>

<style> body {

font-family: 'Gill Sans', Calibri, 'Trebuchet MS', sans-serif; background-color: #E2DFD0;

color: #524C42; margin: 0; height: 100vh; position: relative;

}

.login-container { background-color: #fff; padding: 40px 50px; border-radius: 16px;

box-shadow: 0 6px 16px rgba(0, 0, 0, 0.15); text-align: center;

width: 320px; position: absolute; top: 50%;

left: 50%;

transform: translate(-50%, -50%);

}

.login-container h2 { margin-bottom: 30px; color: #32012F;

}

input[type="text"], input[type="password"] { width: 100%;

padding: 14px; margin-bottom: 20px;

border: 1px solid #ccc; border-radius: 10px; font-size: 16px;

text-align: center;

box-sizing: border-box;

}

input[type="submit"] { background-color: #32012F; color: #E2DFD0;

padding: 14px 25px; border: none; border-radius: 8px; cursor: pointer;

font-size: 16px;

transition: background-color 0.3s; width: 100%;

}

input[type="submit"]:hover { background-color: #F97300;

}

.signup-link { margin-top: 20px; font-size: 14px;

}

.signup-link a { color: #F97300;

text-decoration: none; font-weight: bold;

}

.signup-link a:hover {

text-decoration: underline;

}

</style>

</head>

<body>

<div class="login-container">

<script>

const params = new URLSearchParams(window.location.search); if (params.has('msg')) {

alert(params.get('msg'));

}

</script>

<h2>Admin Login</h2>

<form action="/admin\_login" method="POST">

<input type="text" name="adminID" placeholder="Admin ID" required />

<input type="password" name="password" placeholder="Password" required />

<input type="submit" value="Login" />

</form>

<div class="signup-link">

New Admin? <a href="/admin\_signup.html">Sign Up</a>

</div>

</div>

</body>

</html>

admin\_signup.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0"/>

<title>Admin Signup</title>

<style> body {

font-family: 'Gill Sans', Calibri, 'Trebuchet MS', sans-serif; background-color: #E2DFD0;

color: #524C42; margin: 0; height: 100vh; position: relative;

}

.signup-container { background-color: #fff; padding: 40px 50px; border-radius: 16px;

box-shadow: 0 6px 16px rgba(0, 0, 0, 0.15); text-align: center;

width: 320px; position: absolute; top: 50%;

left: 50%;

transform: translate(-50%, -50%);

}

.signup-container h2 { margin-bottom: 30px; color: #32012F;

}

input {

width: 100%; padding: 14px; margin-bottom: 20px;

border: 1px solid #ccc; border-radius: 10px; font-size: 16px;

text-align: center;

box-sizing: border-box;

}

input[type="submit"] { background-color: #32012F; color: #E2DFD0;

cursor: pointer;

transition: background-color 0.3s;

}

input[type="submit"]:hover { background-color: #F97300;

}

.login-link {

margin-top: 20px; font-size: 14px;

}

.login-link a { color: #F97300;

text-decoration: none; font-weight: bold;

}

.login-link a:hover {

text-decoration: underline;

}

</style>

</head>

<body>

<div class="signup-container">

<h2>Admin Signup</h2>

<form id="signupForm">

<input type="text" name="adminID" placeholder="Admin ID" required />

<input type="password" name="password" placeholder="Password" required />

<input type="password" name="confirmPassword" placeholder="Confirm Password" required />

<input type="text" name="name" placeholder="Name" required />

<input type="submit" value="Sign Up" />

</form>

<div class="login-link">

Already have an account? <a href="admin\_login.html">Login</a>

</div>

</div>

<script>

document.getElementById('signupForm').addEventListener('submit', async function (e) { e.preventDefault();

const form = e.target;

const formData = new FormData(form); try {

const response = await fetch('/admin\_signup', { method: 'POST',

body: formData,

});

if (response.ok) {

alert('Signed up successfully!'); window.location.href = 'admin\_login.html';

} else {

const errorText = await response.text(); alert('Error: ' + errorText);

}

} catch (error) { console.error(error);

alert('An unexpected error occurred.');

}

});

</script

</body>

</html>

delete\_equipment.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0"/>

<title>Delete Equipment</title>

<style> body {

font-family: 'Gill Sans', Calibri, 'Trebuchet MS', sans-serif; background-color: #E2DFD0;

margin: 0;

padding: 0; display: flex;

align-items: center; justify-content: center; height: 100vh;

}

.container {

background-color: #fff; padding: 40px;

border-radius: 12px;

box-shadow: 0 6px 18px rgba(0, 0, 0, 0.2); max-width: 600px;

width: 100%;

}

h2 {

text-align: center; color: #32012F; margin-bottom: 30px;

}

.equipment-list { margin-bottom: 30px;

}

.equipment-item { background-color: #f8f8f8; padding: 15px;

margin-bottom: 15px; border-radius: 8px; display: flex;

justify-content: space-between; align-items: center;

}

.equipment-item h3 { color: #524C42; font-size: 18px;

}

.equipment-item p { color: #524C42; font-size: 14px;

}

.delete-btn {

background-color: #F97300; color: white;

border: none; padding: 10px; border-radius: 6px; cursor: pointer; font-size: 16px;

transition: background-color 0.3s ease;

}

.delete-btn:hover { background-color: #D45D00;

}

.back-link { display: block; text-align: center; margin-top: 20px; color: #32012F;

text-decoration: none; font-weight: bold;

}

.back-link:hover {

text-decoration: underline;

}

</style>

</head>

<body>

<div class="container">

<h2>Delete Equipment</h2>

<div class="equipment-list">

{% for equipment in equipment\_list %}

<div class="equipment-item">

<div>

<h3>{{ equipment[1] }}</h3>

<p>{{ equipment[2] }}</p>

</div>

<form action="/delete\_equipment" method="POST">

<input type="hidden" name="equipment\_id" value="{{ equipment[0] }}">

<button type="submit" class="delete-btn">−</button>

</form>

</div>

{% endfor %}

</div>

<a class="back-link" href="{{ url\_for('admin\_dashboard') }}">← Back to Dashboard</a>

</div>

</body>

</html>

edit-equipment\_html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0"/>

<title>Edit Equipment</title>

<style>

body { margin: 0; font-family: 'Gill Sans', Calibri, 'Trebuchet MS', sans-serif; background-color: #E2DFD0; color: #524C42; }

header { background-color: #32012F; padding: 25px; text-align: center; color: #E2DFD0; font-size: 32px; font-weight: bold; box-shadow: 0 4px 10px rgba(0, 0, 0, 0.2); }

.container { max-width: 900px; margin: 40px auto; padding: 20px; } h2 { color: #32012F; font-size: 28px; margin-bottom: 30px; }

form { background-color: #fff; padding: 25px; border-radius: 12px; box-shadow: 0 6px 16px rgba(0, 0, 0, 0.1); }

label { font-size: 18px; color: #32012F; display: block; margin-bottom: 10px; } input[type="text"], input[type="number"], select { width: 100%; padding: 10px; font-size: 16px;

border: 1px solid #ccc; border-radius: 8px; margin-bottom: 20px; }

button { background-color: #32012F; color: #E2DFD0; padding: 12px 25px; border: none; border-radius: 12px; font-size: 18px; cursor: pointer; transition: background-color 0.3s; }

button:hover { background-color: #F97300; }

a { text-decoration: none; display: inline-block; margin-top: 30px; padding: 12px 25px; background-color: #32012F; color: #E2DFD0; border-radius: 12px; font-size: 18px; transition: background-color 0.3s; }

a:hover { background-color: #F97300; }

</style>

</head>

<body>

<header> Edit Equipment - Admin Dashboard </header>

<div class="container">

<h2>Edit Equipment</h2>

<form method="POST">

<label for="equipmentName">Equipment Name:</label>

<input type="text" id="equipmentName" name="equipmentName" value="{{ equipment[1] }}" required>

<label for="description">Description:</label>

<input type="text" id="description" name="description" value="{{ equipment[2] }}" required>

<label for="quantity">Quantity:</label>

<input type="number" id="quantity" name="quantity" value="{{ equipment[3] }}" required>

<label for="category">Category:</label>

<input type="text" id="category" name="category" value="{{ equipment[4] }}" required>

<label for="status">Status:</label>

<select name="status" id="status" required>

<option value="Available" {% if equipment[5] == 'Available' %}selected{% endif

%}>Available</option>

<option value="Unavailable" {% if equipment[5] == 'Unavailable' %}selected{% endif

%}>Unavailable</option>

</select>

<button type="submit">Update Equipment</button>

</form>

<a href="{{ url\_for('admin\_dashboard') }}">Back to Dashboard</a>

</div>

</body>

</html>

update-equipment.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0"/>

<title>Update Equipment</title>

<style>

body { margin: 0; font-family: 'Gill Sans', Calibri, 'Trebuchet MS', sans-serif; background-color: #E2DFD0; color: #524C42; }

header { background-color: #32012F; padding: 25px; text-align: center; color: #E2DFD0; font-size: 32px; font-weight: bold; box-shadow: 0 4px 10px rgba(0, 0, 0, 0.2); }

.container { max-width: 1200px; margin: 40px auto; padding: 20px; }

.equipment-list { list-style-type: none; padding: 0; margin-bottom: 50px; }

.equipment-item { background-color: #fff; margin: 10px 0; padding: 20px; border-radius: 12px; box-shadow: 0 6px 16px rgba(0, 0, 0, 0.1); display: flex; justify-content: space-between; align-items: center; }

.equipment-details { font-size: 18px; color: #32012F; }

.equipment-actions a { text-decoration: none; padding: 8px 16px; background-color: #32012F; color: #E2DFD0; border-radius: 12px; transition: background-color 0.3s; }

.equipment-actions a:hover { background-color: #F97300; }

.back-btn { display: inline-block; margin-top: 30px; padding: 12px 25px; background-color: #32012F; color: #E2DFD0; border-radius: 12px; text-decoration: none; font-size: 18px; text-align: center; transition: background-color 0.3s; }

.back-btn:hover { background-color: #F97300; }

</style>

</head>

<body>

<header> Update Equipment - Admin Dashboard </header>

<div class="container">

<h2>Update Equipment</h2>

<ul class="equipment-list">

{% for equipment in equipment\_list %}

<li class="equipment-item">

<div class="equipment-details">

<strong>{{ equipment[1] }}</strong><br>

{{ equipment[2] }}

</div>

<div class="equipment-actions">

<a href="{{ url\_for('edit\_equipment', equipment\_id=equipment[0]) }}">Edit</a>

</div>

</li>

{% endfor %}

</ul>

<a href="{{ url\_for('admin\_dashboard') }}" class="back-btn">Back to Dashboard</a>

</div>

</body>

</html>

user\_login.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>User Login</title>

<style>

body {

font-family: 'Gill Sans', Calibri, 'Trebuchet MS', sans-serif; background-color: #E2DFD0;

color: #524C42; margin: 0; height: 100vh; position: relative;

}

.login-container { background-color: #fff; padding: 40px 50px; border-radius: 16px;

box-shadow: 0 6px 16px rgba(0, 0, 0, 0.15); text-align: center;

width: 320px; position: absolute; top: 50%;

left: 50%;

transform: translate(-50%, -50%);

}

.login-container h2 { margin-bottom: 30px; color: #32012F;

}

input, select { width: 100%; padding: 14px;

margin-bottom: 20px; border: 1px solid #ccc;

border-radius: 10px; font-size: 16px;

text-align: center;

box-sizing: border-box;

}

input[type="submit"] { background-color: #32012F; color: #E2DFD0;

cursor: pointer;

transition: background-color 0.3s;

}

input[type="submit"]:hover { background-color: #F97300;

}

.signup-link { margin-top: 20px; font-size: 14px;

}

.signup-link a { color: #F97300;

text-decoration: none; font-weight: bold;

}

.signup-link a:hover {

text-decoration: underline;

}

</style>

</head>

<body>

<div class="login-container">

<h2>User Login</h2>

<form action="/user\_login" method="POST">

<input type="email" name="email" placeholder="Email" required />

<input type="password" name="password" placeholder="Password" required />

<input type="submit" value="Login" />

</form>

<div class="signup-link">

Don't have an account? <a href="/user\_signup">Sign up</a>

</div>

</div>

</body>

</html>

user\_signup.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>User Signup</title>

<style> body {

font-family: 'Gill Sans', Calibri, 'Trebuchet MS', sans-serif; background-color: #E2DFD0;

color: #524C42;

margin: 0; height: 100vh; position: relative;

}

.signup-container { background-color: #fff; padding: 40px 50px; border-radius: 16px;

box-shadow: 0 6px 16px rgba(0, 0, 0, 0.15); text-align: center;

width: 320px; position: absolute; top: 50%;

left: 50%;

transform: translate(-50%, -50%);

}

.signup-container h2 { margin-bottom: 30px; color: #32012F;

}

input, select { width: 100%; padding: 14px;

margin-bottom: 20px; border: 1px solid #ccc; border-radius: 10px; font-size: 16px;

text-align: center;

box-sizing: border-box;

}

input[type="submit"] { background-color: #32012F; color: #E2DFD0;

cursor: pointer;

transition: background-color 0.3s;

}

input[type="submit"]:hover { background-color: #F97300;

}

.login-link {

margin-top: 20px; font-size: 14px;

}

.login-link a { color: #F97300;

text-decoration: none; font-weight: bold;

}

.login-link a:hover {

text-decoration: underline;

}

</style>

</head>

<body>

<div class="signup-container">

<h2>User Signup</h2>

<form action="/user\_signup" method="POST">

<input type="text" name="name" placeholder="Name" required />

<input type="email" name="email" placeholder="Email" required />

<input type="password" name="password" placeholder="Password" required />

<input type="password" name="confirmPassword" placeholder="Confirm Password" required />

<input type="tel" name="phone" placeholder="Phone Number" required />

<select name="gender" required>

<option value="" disabled selected>Select Gender</option>

<option value="female">Female</option>

<option value="male">Male</option>

<option value="other">Other</option>

</select>

<input type="submit" value="Sign Up" />

</form>

<div class="login-link">

Already have an account? <a href="user\_login.html">Login</a>

</div>

</div>

</body>

</html>

user-dashboard.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0"/>

<title>User Dashboard</title>

<style> body { margin: 0;

font-family: 'Helvetica Neue', sans-serif; background-color: #f4f7fb;

color: #333;

line-height: 1.6;

}

header {

background-color: #32012F; padding: 40px;

text-align: center; color: #ffffff;

font-size: 36px; font-weight: bold;

box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);

}

.notification {

background-color: #f3f3f3; padding: 20px;

text-align: center; font-size: 18px; color: #444;

border-bottom: 2px solid #ccc; margin-top: 20px;

font-weight: 500;

}

.categories { display: flex;

justify-content: center; margin: 30px 0;

flex-wrap: wrap; gap: 25px;

}

.category-btn {

background-color: #32012F; color: #ffffff;

border: none; padding: 15px 30px; font-size: 18px; border-radius: 12px; cursor: pointer;

transition: background-color 0.3s ease, transform 0.3s ease; box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);

}

.category-btn:hover,

.category-btn.active { background-color: #F97300; transform: translateY(-3px);

}

.container {

max-width: 1200px; margin: 0 auto; padding: 30px;

}

.grid {

display: none;

grid-template-columns: repeat(auto-fill, minmax(300px, 1fr)); gap: 20px;

margin-top: 20px;

}

.grid.active { display: grid;

}

.card {

background-color: #ffffff; border-radius: 10px;

box-shadow: 0 4px 12px rgba(0, 0, 0, 0.1); overflow: hidden;

transition: transform 0.3s ease, box-shadow 0.3s ease;

}

.card:hover {

transform: translateY(-5px);

box-shadow: 0 6px 15px rgba(0, 0, 0, 0.15);

}

.card-content { padding: 20px;

}

.card-content h3 { margin: 0 0 15px; font-size: 22px;

color: #32012F;

}

.card-content p { font-size: 16px; margin: 5px 0;

}

.rent {

color: #F97300; font-weight: bold;

}

.availability {

font-weight: bold;

}

.availability.available { color: green;

}

.availability.unavailable { color: red;

}

.book-btn {

margin-top: 20px; width: 100%;

background-color: #32012F; color: white;

border: none; padding: 15px; border-radius: 10px; cursor: pointer;

font-size: 18px;

transition: background-color 0.3s ease, transform 0.3s ease;

}

.book-btn:hover { background-color: #F97300; transform: translateY(-3px);

}

.book-btn:disabled { background-color: #ccc; cursor: not-allowed;

}

@media (max-width: 768px) { header { font-size: 28px; }

.notification { font-size: 16px; }

.category-btn { padding: 12px 25px; font-size: 16px; }

.card-content h3 { font-size: 20px; }

.book-btn { font-size: 16px; }

}

@media (max-width: 480px) { header { font-size: 24px; }

.category-btn { padding: 10px 20px; font-size: 14px; }

.card-content h3 { font-size: 18px; }

.book-btn { font-size: 14px; }

}

</style>

</head>

<body>

<header>

User Dashboard - Equipment Booking

</header>

<div class="notification">

Welcome, <span id="user-name">{{ username }}</span>! Select a category to see available equipment.

</div>

<div class="categories">

<button class="category-btn" onclick="filterCategory('agriculture')">Agriculture</button>

<button class="category-btn" onclick="filterCategory('construction')">Construction</button>

<button class="category-btn" onclick="filterCategory('industrial')">Industrial</button>

<button class="category-btn" onclick="filterCategory('transportation')">Transportation</button>

<button class="category-btn" onclick="filterCategory('all')">All</button>

</div>

<div class="container">

{% set categories = ['agriculture', 'construction', 'industrial', 'transportation'] %}

{% for cat in categories %}

<div class="grid {{ cat }} {% if selected\_category == cat %}active{% endif %}">

{% for eq in equipment\_list %}

{% if eq[4]|lower == cat %}

<div class="card">

<div class="card-content">

<h3>{{ eq[1] }}</h3>

<p>{{ eq[2] }}</p>

<p class="rent">Quantity: {{ eq[3] }}</p>

<p class="availability {{ 'available' if eq[5]|lower == 'available' else 'unavailable' }}">{{ eq[5]

}}</p>

<button class="book-btn" onclick="showConfirmation(event, '{{ eq[0] }}', '{{ user\_id }}')" {% if

eq[5]|lower != 'available' %}disabled{% endif %}>Book Now</button>

</div>

</div>

{% endif %}

{% endfor %}

</div>

{% endfor %}

<div class="grid all {% if selected\_category == 'all' %}active{% endif %}">

{% for eq in equipment\_list %}

<div class="card">

<div class="card-content">

<h3>{{ eq[1] }}</h3>

<p>{{ eq[2] }}</p>

<p class="rent">Quantity: {{ eq[3] }}</p>

<p class="availability {{ 'available' if eq[5]|lower == 'available' else 'unavailable' }}">{{ eq[5]

}}</p>

<button class="book-btn" onclick="showConfirmation(event, '{{ eq[0] }}', '{{ user\_id }}')" {% if eq[5]|lower != 'available' %}disabled{% endif %}>Book Now</button>

</div>

</div>

{% endfor %}

</div>

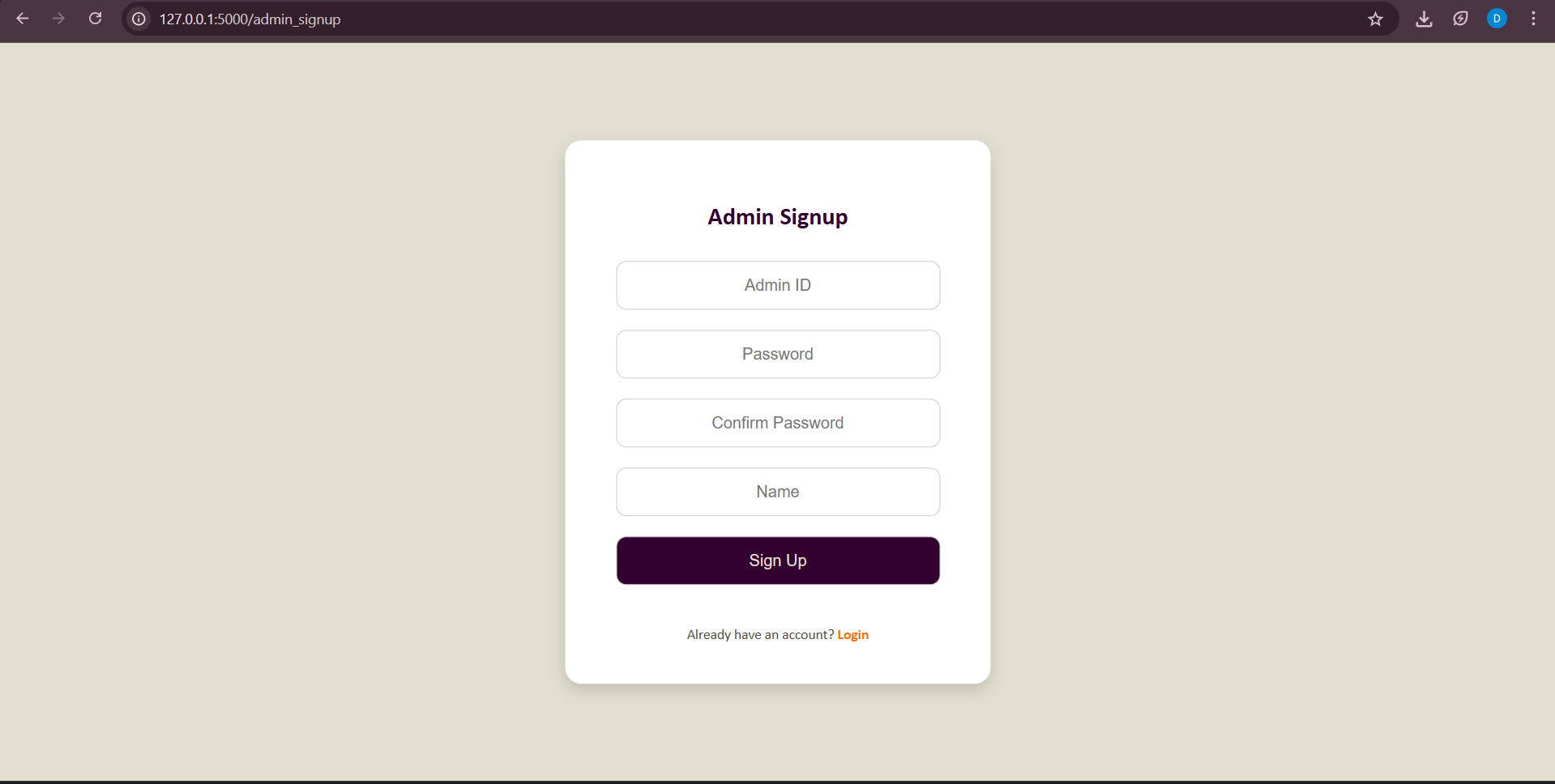
</div>

<script>

function filterCategory(category) {

document.querySelectorAll('.grid').forEach(grid => grid.classList.remove('active')); const selectedGrid = document.querySelector(`.grid.${category}`);

if (selectedGrid) selectedGrid.classList.add('active');



}

window.onload = () => {

const selected = "{{ selected\_category }}"; if (selected) filterCategory(selected);

else filterCategory("all");

}

function showConfirmation(event, equipmentId, userId) {

if (confirm("Are you sure you want to book this equipment?")) { const formData = new FormData(); formData.append('equipment\_id', equipmentId); formData.append('user\_id', userId);

fetch('/book\_equipment', { method: 'POST',

body: formData

})

.then(response => response.text())

.then(data => {

alert("Booking request submitted successfully!"); location.reload();

})

.catch(error => {

alert("Error occurred while booking. Please try again."); console.error("Error booking equipment:", error);

});

}

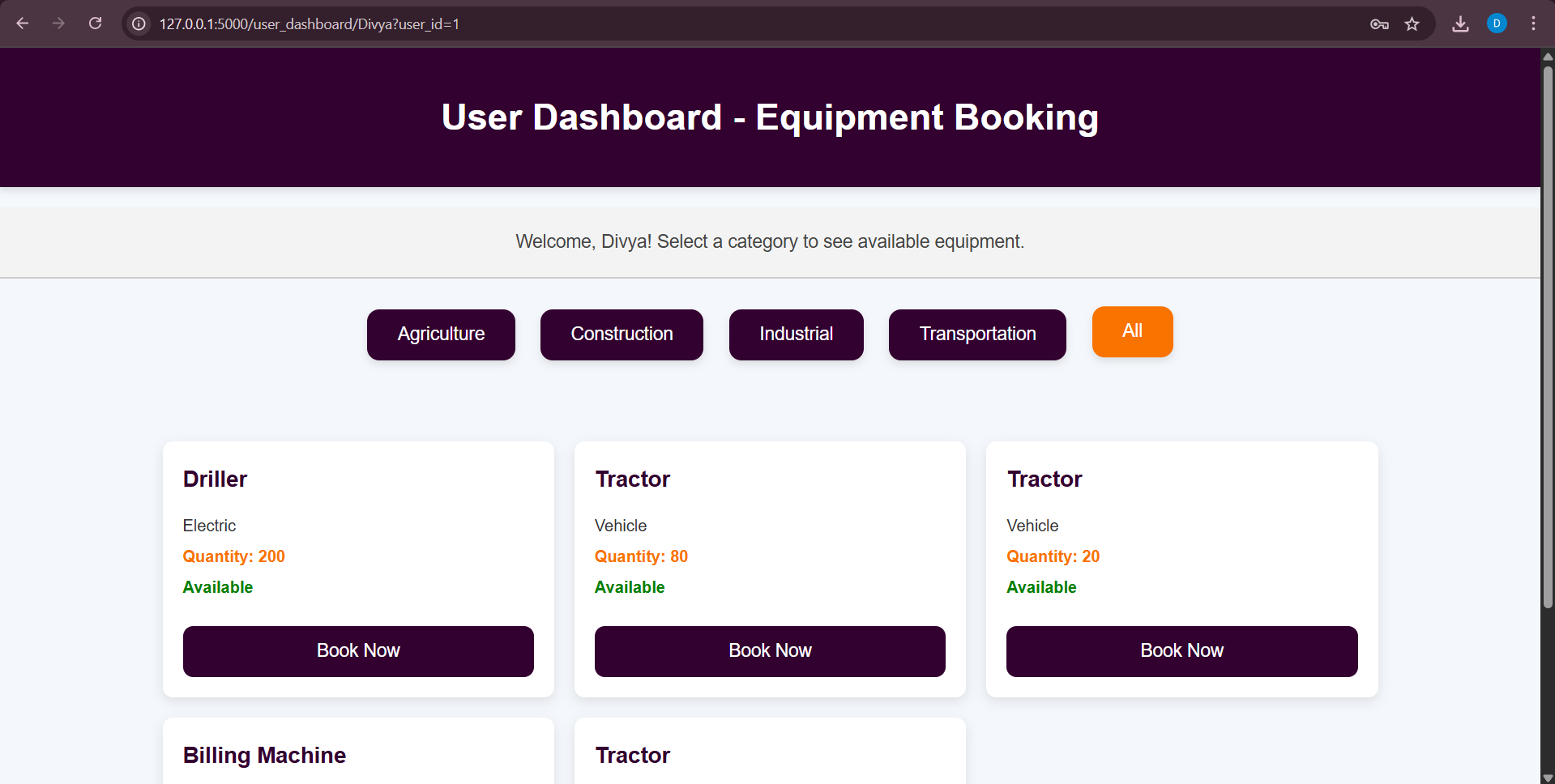
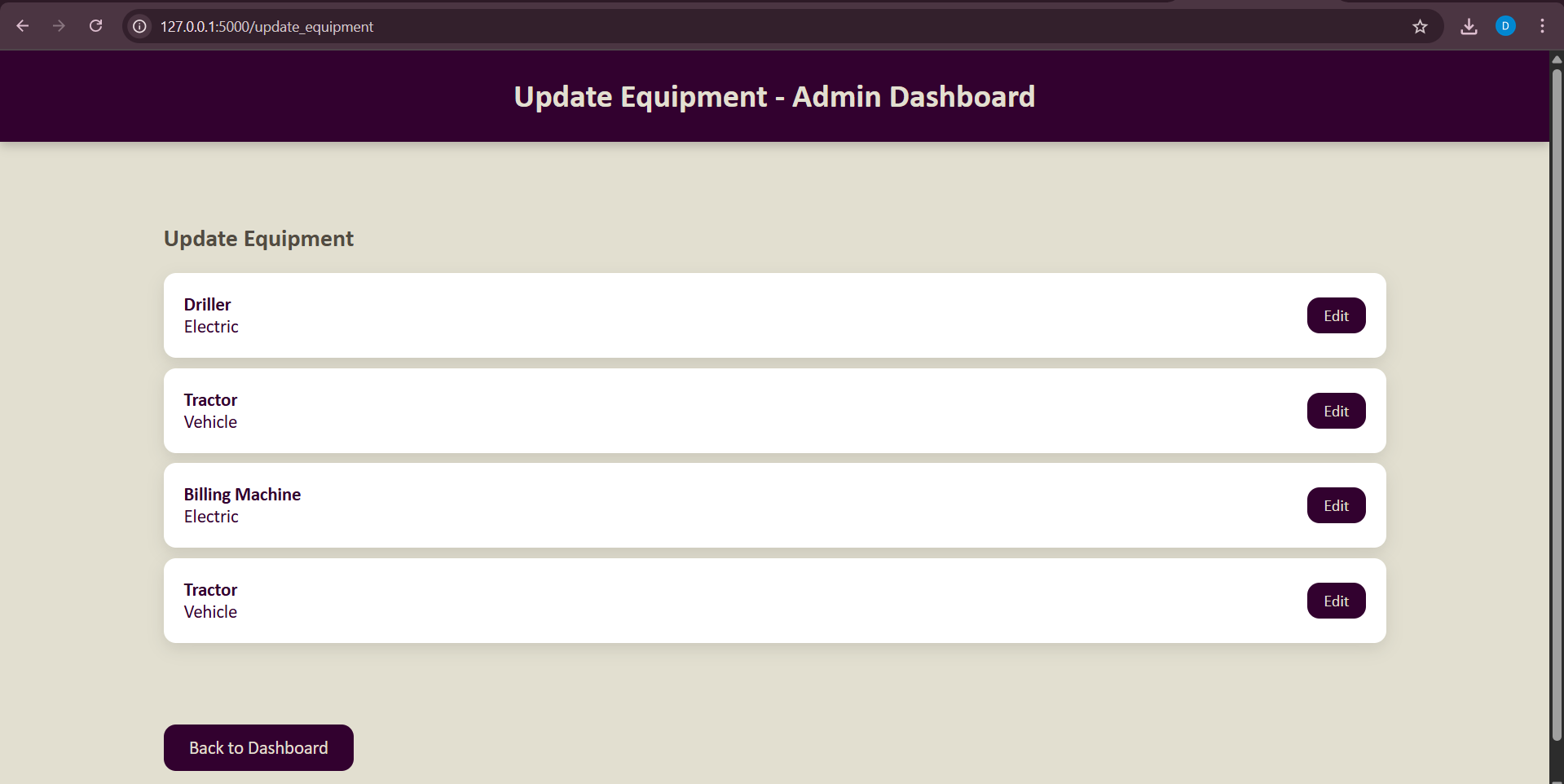
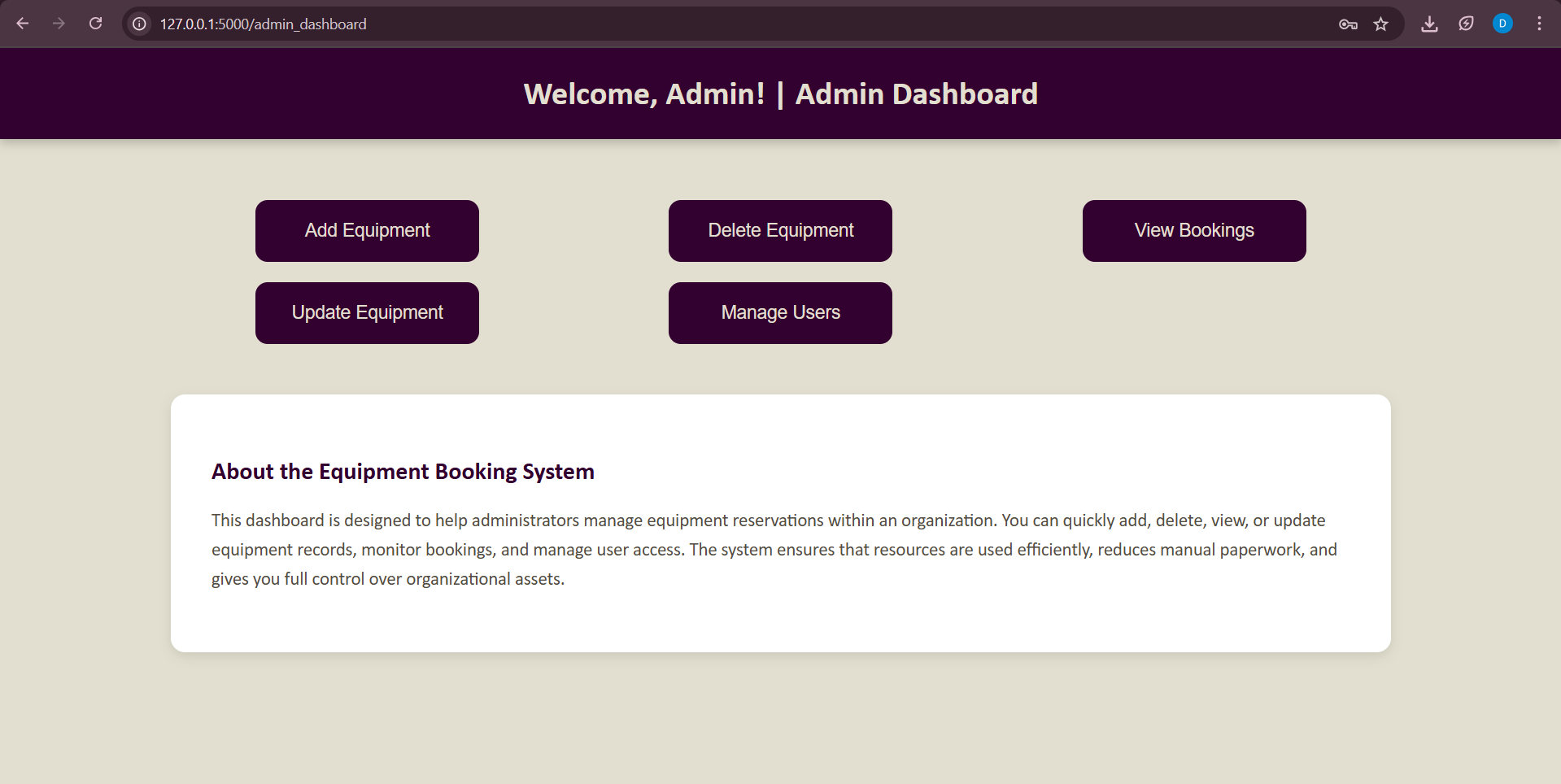
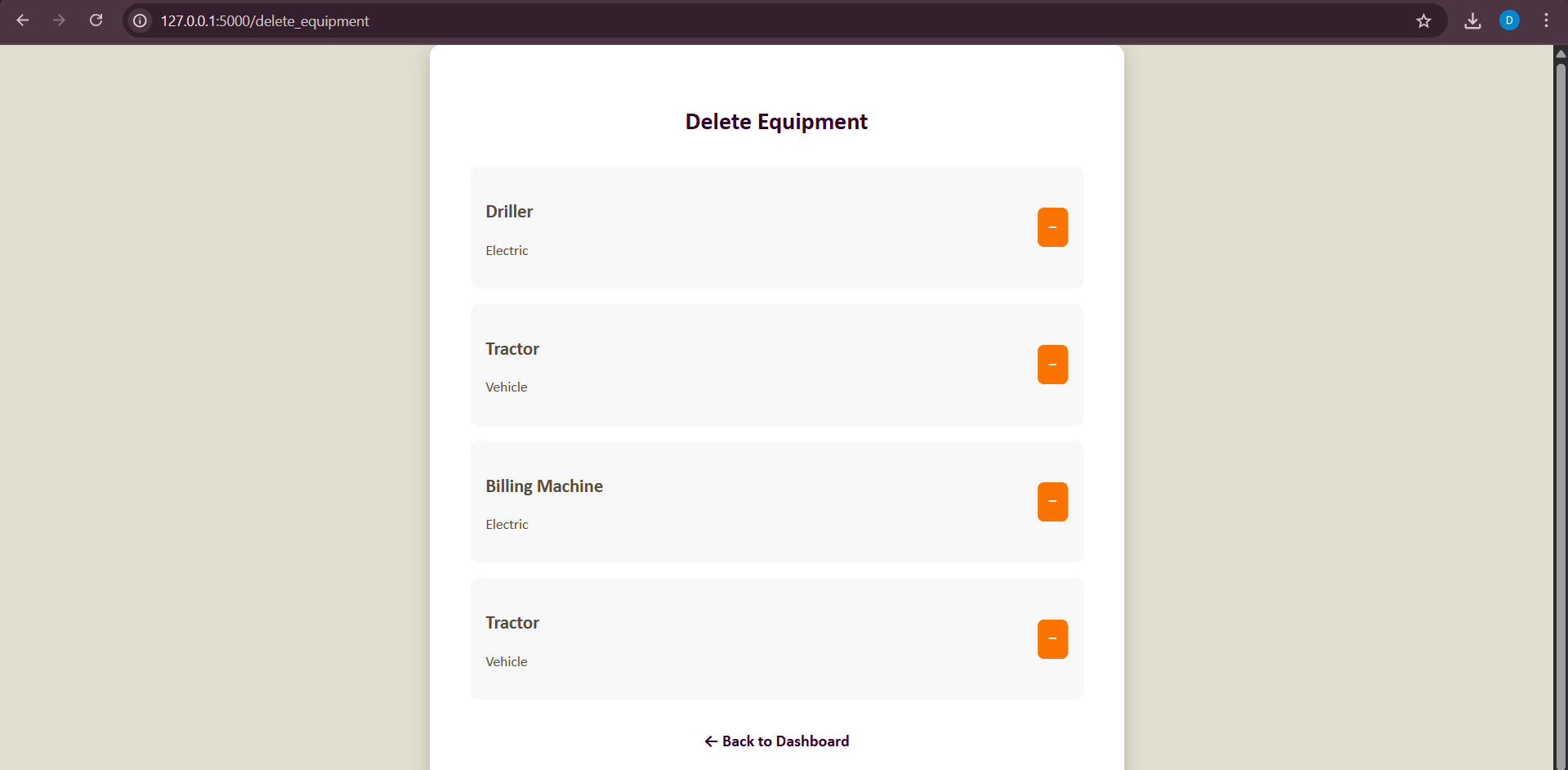
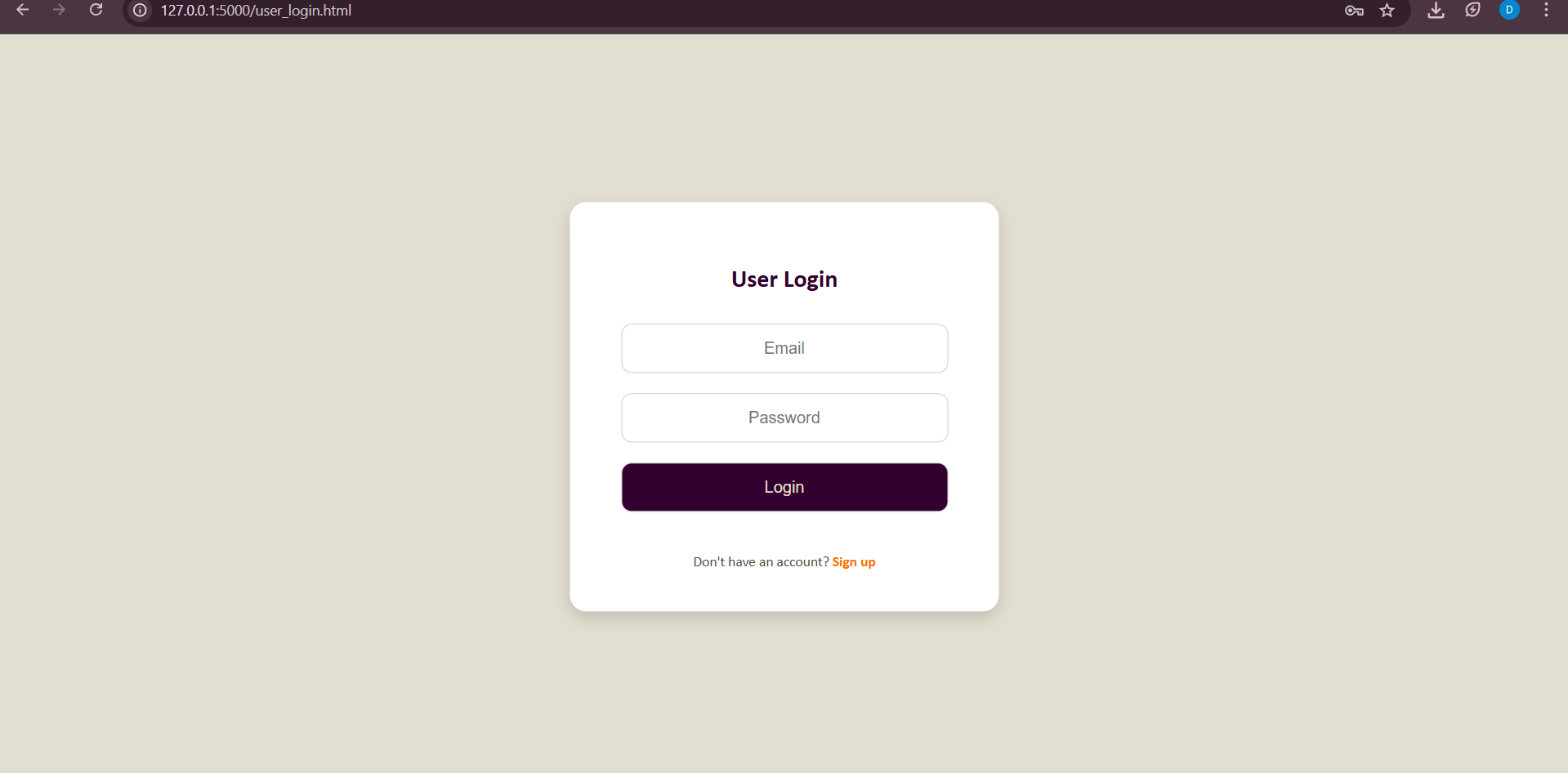
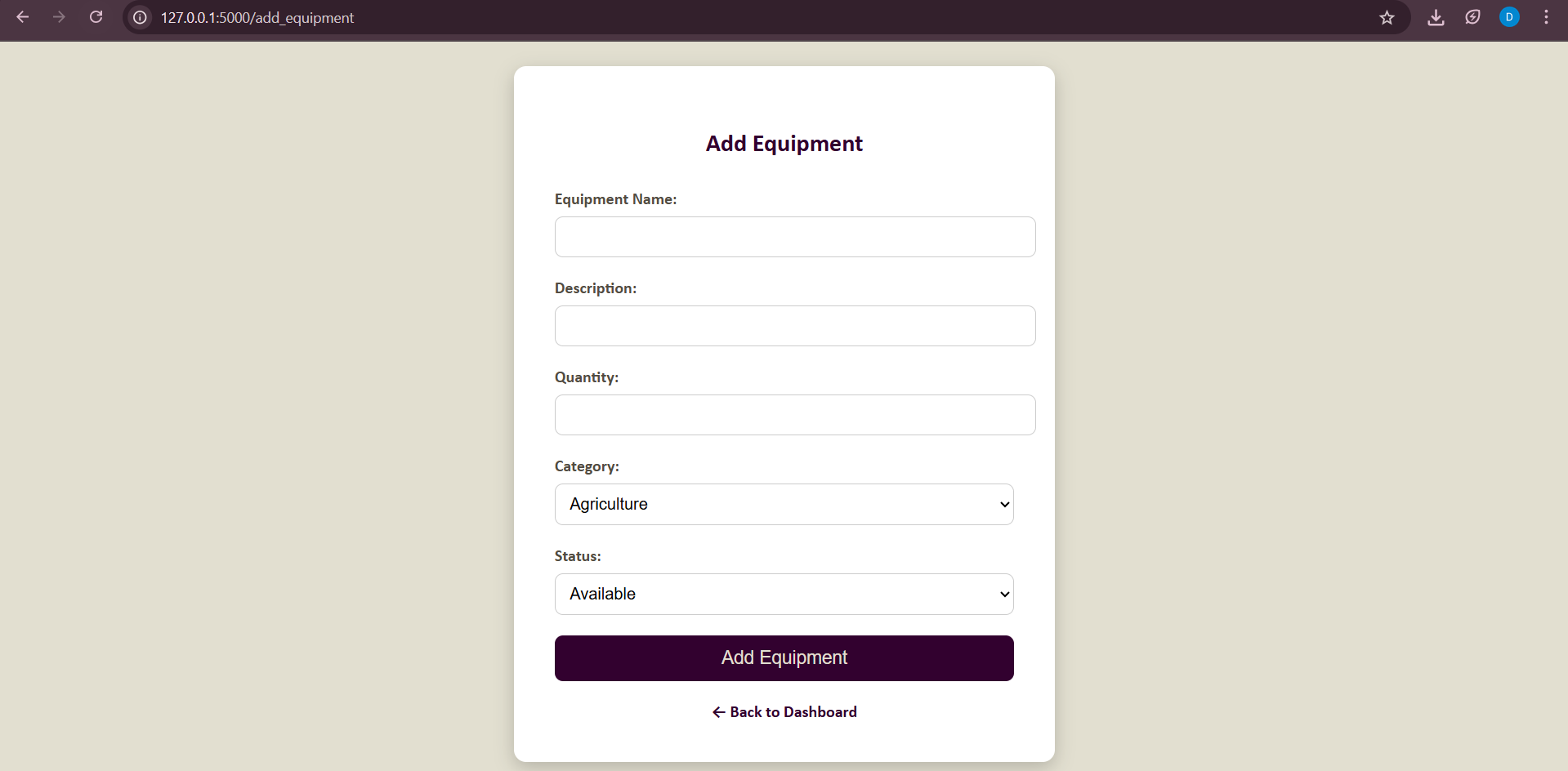
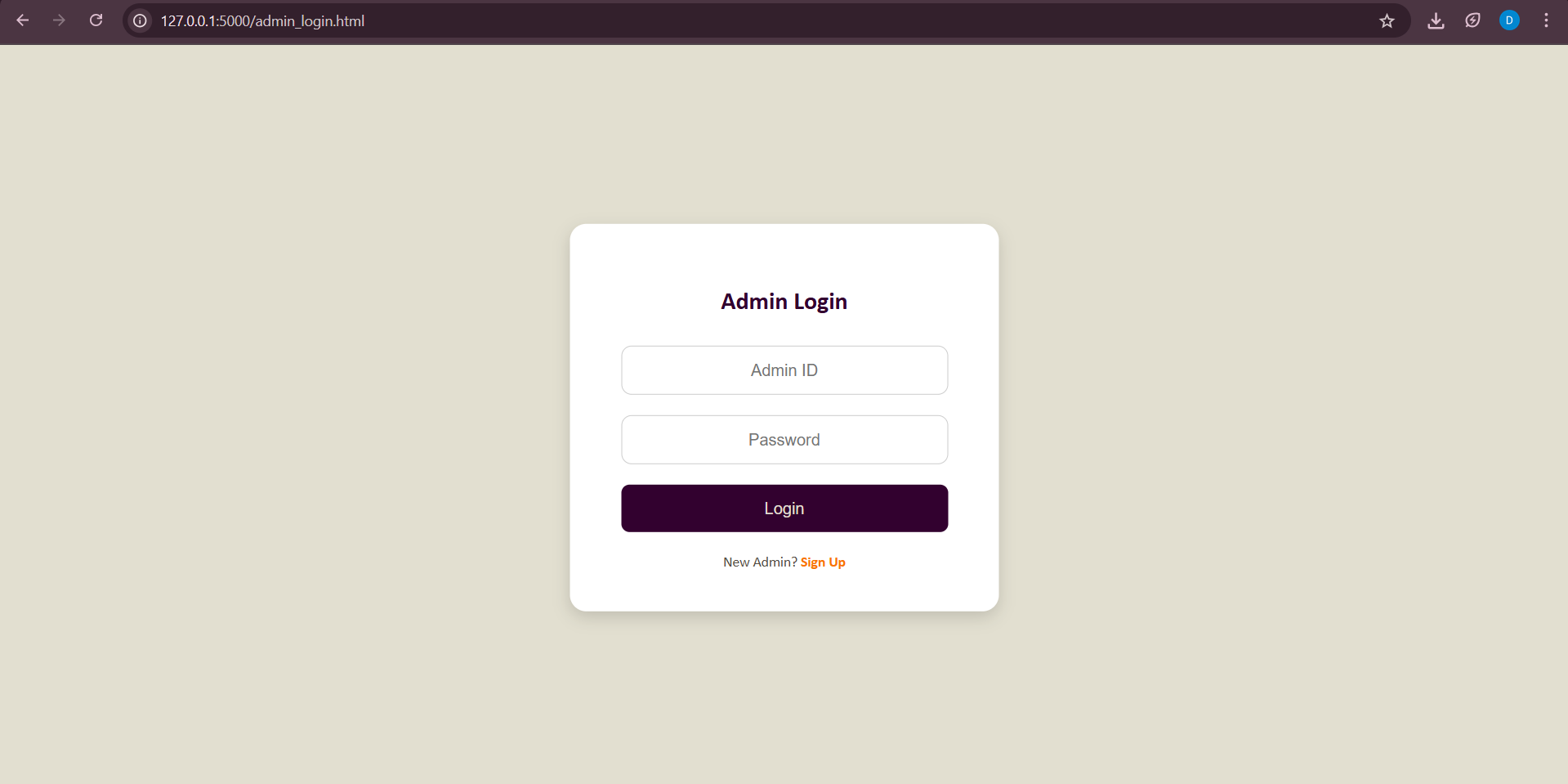
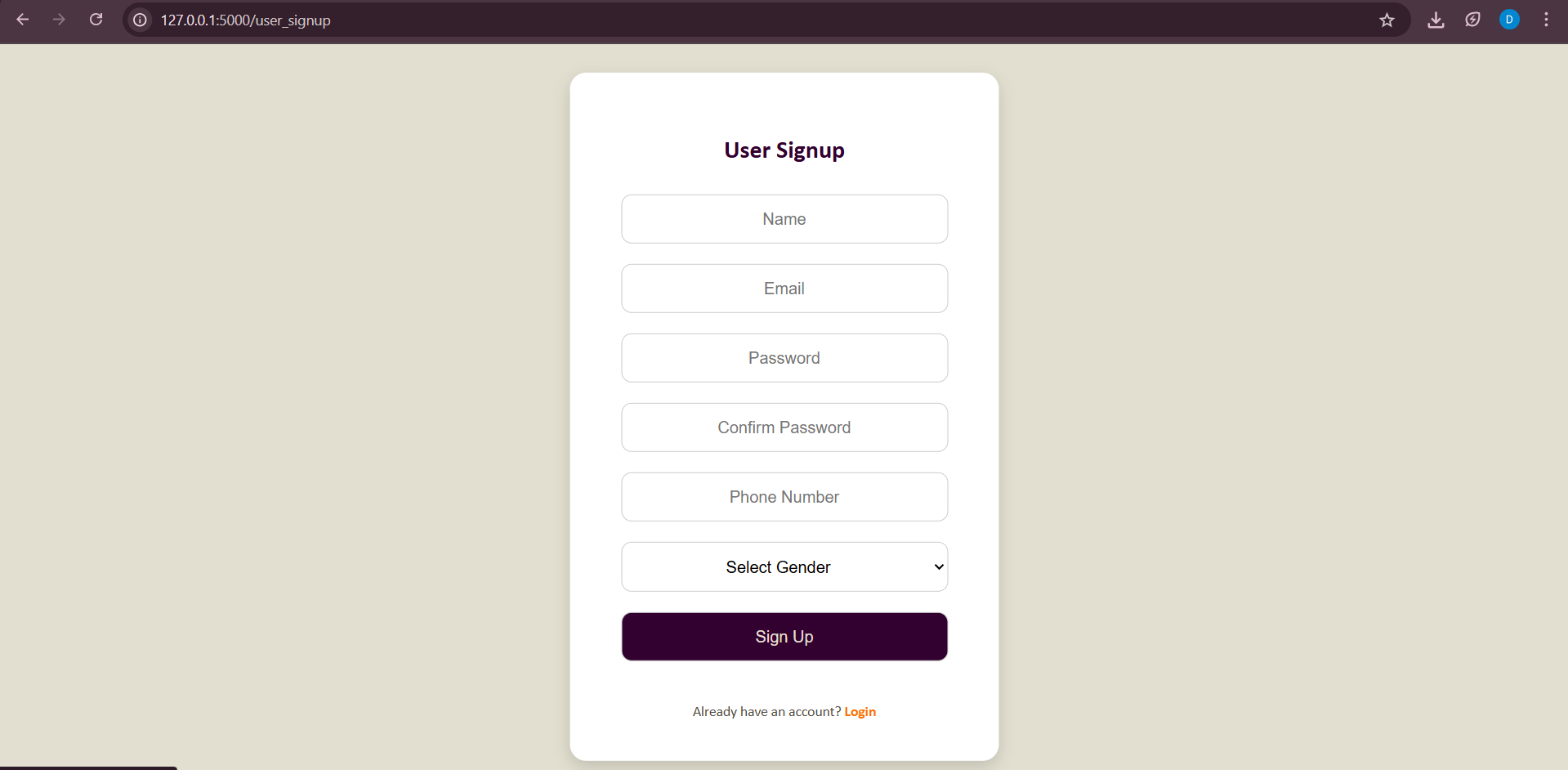
}

</script>

</body>

</html>

# Output:



**Reference:**

* **R. Elmasri and S. B. Navathe**, *Fundamentals of Database Systems*, 7th ed., Pearson, 2017.
* <https://dev.mysql.com/doc/>
* <https://sqlbolt.com/>
* https://mode.com/sql-tutorial/