

**MICRO PROJECT**  
**E-GOVERNANCE SYSTEM**

**BY**  
**JAYRAGHAV V**  
**JOHN BELLARMINE**

## INDEX

S.NO	TOPIC	PAGE NO
1	PROBLEM STATEMENT	3
2	REQUIREMENTS	4
3	DESIGN	6
4	IMPLEMENTATION	11
5	SCREENSHOTS	32
6	CONCLUSION	36

## **PROBLEM STATEMENT:**

In a comprehensive e-governance system, the profit or income of the country from various sources will be divided and allotted to different departments/ministries based on the budget and requirements.

Every department/ministry has its own set of employees, and their details are maintained in their respective departments. Employees have their own login credentials, their respective roles or positions in the department and permission/access to certain works.

The employees also look after the bill generation process and payment pathways through which services are provided to the public. These services are listed down and managed in the common service centers called CSC.

Every government function with the help of the public. Every citizen plays a major role. So, the government maintains the details of the citizens and provides them with unique login credentials to access the websites and services provided by the government. So, all the citizens including the employees have unique login credentials.

With the help of these login credentials, the citizens of the country access the services provided, Public Distribution System (PDS), Common Service Centers (CSC) details and many more. The government is also responsible for the employment of its citizens, so they also update the citizens with the vacancies available in the companies which are enrolled under the government.

The companies enrolled under the government are provided with projects. And the records of the past and ongoing projects are maintained by the companies and government so that the citizens would be aware about the funding of the government for the projects. Having all these sections would increase the efficiency of the relationship between the government and the citizens

## Requirements

### Overview: E-Governance Portal

This document details the specifications for the E-Governance Portal, a web-based application developed using the Flask framework. The portal serves as an interactive interface for citizens and government employees, facilitating smooth access to and management of a variety of governmental services. It includes detailed descriptions of the portal's core functionalities, data management capabilities, and areas identified for potential future enhancements.

### Core Functionalities

The E-Governance Portal is built to ensure secure, user-specific interactions, where functionality is tailored to the role of the user, be it an employee or a citizen.

#### 1. User Authentication

To ensure secure access:

- **User Types:** The portal supports two user roles: employees and citizens.
- **Login Process:** Upon entering the site, users are guided to a login page to select their role and input credentials.
- **Authentication:** User credentials are verified against a MySQL database. Successful login grants access to role-specific dashboards, with session management mechanisms ensuring a seamless user experience.
- **Session Management:** This maintains user authentication status throughout their session, allowing access to appropriate features while preventing unauthorized access.

#### 2. Dashboard Features

Two distinct dashboards cater to the needs of employees and citizens, offering tailored features:

- **Employee Dashboard:** Employees can access tools to manage government service information. Core functionalities include:
  - **Permissions Management:** Employees are assigned permissions that define their access levels within the portal.
  - **CRUD Operations:** Employees can perform Create, Read, Update, and Delete (CRUD) operations on key data entities, including Services, Departments, and Common Service Centres (CSCs).
- **Citizen Dashboard:** Citizens are offered a range of accessible services, including:
  - **Service Listings:** Details about available government services, fees, and availability are displayed.
  - **Job Vacancies:** Citizens can view current job openings within government sectors.
  - **CSC Information:** Location and contact details for nearby CSCs are provided.

### 3. Data Management

The data management functionality equips employees with a streamlined interface to interact with various data tables:

- **View Records:** Employees can view comprehensive listings within a selected database table.
- **Insert Records:** A straightforward form-based entry system allows for adding new records to the database.
- **Update Records:** Existing records can be modified, ensuring up-to-date information.
- **Delete Records:** Obsolete records can be easily removed from the system.

These operations are facilitated through RESTful Flask routes, handling HTTP methods (GET, POST, PUT, DELETE) to maintain a robust backend structure.

### Additional Features

The portal incorporates the following enhancements to create a seamless user experience:

- **Error Handling:** Intuitive flash messages keep users informed about successful actions and errors, such as invalid login attempts.
- **Responsive Design:** Tailwind CSS ensures that the portal's interface is fully responsive, providing a modern user experience across various devices.

Looking ahead, the E-Governance Portal offers ample opportunities for enhancement:

- **Enhanced Security:** Future updates may include OAuth or JWT-based authentication to further secure user data.
- **Mobile Accessibility:** Expanding to a mobile application would make the portal more accessible to citizens.
- **Service Integration:** Connecting the portal to additional governmental databases could enable real-time data updates.
- **Analytics Dashboard:** Introducing an analytics module for employees could enable tracking of service usage, demand trends, and overall system performance.

The E-Governance Portal serves as an efficient bridge between citizens and government services, with specific tools enabling employees to effectively manage these resources. Each feature has been designed with user needs in mind, creating a user-friendly, responsive, and secure digital platform. The planned future enhancements are aimed at establishing the portal as a key tool in supporting digital governance initiatives.

## DESIGN:

```
mysql> use e_governance;  
Database changed  
mysql> show tables;
```

Tables_in_e_governance
bill
citizen
citizen_documents
citizen_login
citizen_logs
citizen_service_access
companies
csc
csc_schemes
department
employee_department
employees
login
payment
pds
permission
projects
service_csc
services
vacancies

20 rows in set (0.00 sec)

```
mysql> desc citizen;
```

Field	Type	Null	Key	Default	Extra
Aadhar	varchar(12)	NO	PRI	NULL	
Name	varchar(255)	YES		NULL	
DOB	date	YES		NULL	
Gender	varchar(255)	YES		NULL	
role	varchar(255)	YES		NULL	

5 rows in set (0.01 sec)

```
mysql> desc citizen_documents;
```

Field	Type	Null	Key	Default	Extra
doc_id	int(11)	NO	PRI	NULL	
Aadhar	varchar(12)	YES	MUL	NULL	
doc_type	varchar(255)	YES		NULL	
doc_value	text	YES		NULL	

4 rows in set (0.04 sec)

```
mysql> desc citizen_login;
```

Field	Type	Null	Key	Default	Extra
Usname	varchar(255)	NO	PRI	NULL	
pwd	varchar(255)	YES		NULL	
Aadhar	varchar(12)	YES	MUL	NULL	

3 rows in set (0.03 sec)

```
mysql> desc citizen_logs;
```

Field	Type	Null	Key	Default	Extra
log_id	int(11)	NO	PRI	NULL	auto_increment
Usname	varchar(255)	YES	MUL	NULL	
action	varchar(255)	YES		NULL	
timestamp	timestamp	NO		current_timestamp()	

```
4 rows in set (0.02 sec)
```

```
mysql> desc citizen_service_access;
```

Field	Type	Null	Key	Default	Extra
Usname	varchar(255)	NO	PRI	NULL	
serv_code	int(11)	NO	PRI	NULL	

```
2 rows in set (0.05 sec)
```

```
mysql> desc companies;
```

Field	Type	Null	Key	Default	Extra
Company_id	int(11)	NO	PRI	NULL	
name	varchar(255)	YES		NULL	
mail_id	varchar(255)	YES		NULL	
category	varchar(255)	YES		NULL	
sector	varchar(255)	YES		NULL	
Phone_no	varchar(15)	YES		NULL	
address	text	YES		NULL	

```
7 rows in set (0.05 sec)
```

```
mysql> desc csc;
```

Field	Type	Null	Key	Default	Extra
csc_id	int(11)	NO	PRI	NULL	
Phone_no	varchar(15)	YES		NULL	
Dept_id	int(11)	YES	MUL	NULL	
centre_name	varchar(255)	YES		NULL	
location	varchar(255)	YES		NULL	

```
5 rows in set (0.03 sec)
```

```
mysql> desc csc_schemes;
```

Field	Type	Null	Key	Default	Extra
scheme_id	int(11)	NO	PRI	NULL	
scheme	varchar(255)	YES		NULL	
csc_id	int(11)	YES	MUL	NULL	

```
3 rows in set (0.03 sec)
```

```
mysql> desc department;
```

Field	Type	Null	Key	Default	Extra
Dept_id	int(11)	NO	PRI	NULL	
Dept_name	varchar(255)	YES		NULL	
amt_coll	decimal(10,2)	YES		NULL	
coll_method	varchar(255)	YES		NULL	

```
4 rows in set (0.04 sec)
```

```
mysql> desc employee_department;
```

Field	Type	Null	Key	Default	Extra
emp_id	int(11)	NO	PRI	NULL	
Dept_id	int(11)	NO	PRI	NULL	

```
2 rows in set (0.03 sec)
```

```
mysql> desc employees;
```

Field	Type	Null	Key	Default	Extra
emp_id	int(11)	NO	PRI	NULL	
Name	varchar(255)	YES		NULL	
DOB	date	YES		NULL	
Postn	varchar(255)	YES		NULL	
Type	varchar(255)	YES		NULL	
Salary	decimal(10,2)	YES		NULL	
Dept_id	int(11)	YES	MUL	NULL	

```
7 rows in set (0.03 sec)
```

```
mysql> desc login;
```

Field	Type	Null	Key	Default	Extra
Usname	varchar(255)	NO	PRI	NULL	
pwd	varchar(255)	YES		NULL	
emp_id	int(11)	YES	MUL	NULL	

```
3 rows in set (0.03 sec)
```

```
mysql> desc payment;
```

Field	Type	Null	Key	Default	Extra
payment_id	int(11)	NO	PRI	NULL	
Type	varchar(255)	YES		NULL	
source_accno	varchar(255)	YES		NULL	
sink_accno	varchar(255)	YES		NULL	
emp_id	int(11)	YES	MUL	NULL	
serv_code	int(11)	YES	MUL	NULL	

```
6 rows in set (0.03 sec)
```



```
mysql> desc pds;
```

Field	Type	Null	Key	Default	Extra
rationno	int(11)	NO	PRI	NULL	
goods	varchar(255)	YES		NULL	
type	varchar(255)	YES		NULL	
Aadhar	varchar(12)	YES	MUL	NULL	

```
4 rows in set (0.02 sec)
```

```
mysql> desc permission;
```

Field	Type	Null	Key	Default	Extra
permission_id	int(11)	NO	PRI	NULL	
type	varchar(255)	YES		NULL	
emp_id	int(11)	YES	MUL	NULL	

```
3 rows in set (0.03 sec)
```

```
mysql> desc projects;
```

Field	Type	Null	Key	Default	Extra
Project_id	int(11)	NO	PRI	NULL	
Company_id	int(11)	YES	MUL	NULL	
scheme_id	int(11)	YES	MUL	NULL	
amt_allotted	decimal(10,2)	YES		NULL	

```
4 rows in set (0.03 sec)
```

```
mysql> desc service_csc;
```

Field	Type	Null	Key	Default	Extra
serv_code	int(11)	NO	PRI	NULL	
csc_id	int(11)	NO	PRI	NULL	

```
2 rows in set (0.03 sec)
```

```
mysql> desc services;
```

Field	Type	Null	Key	Default	Extra
serv_code	int(11)	NO	PRI	NULL	
Dept_id	int(11)	YES	MUL	NULL	
serv	varchar(255)	YES		NULL	
govt_fee	decimal(10,2)	YES		NULL	

```
4 rows in set (0.03 sec)
```

```
mysql> desc vacancies;
```

Field	Type	Null	Key	Default	Extra
vacancy_id	int(11)	NO	PRI	NULL	
Company_id	int(11)	YES	MUL	NULL	
locality	varchar(255)	YES		NULL	
vacancies	int(11)	YES		NULL	
salary	decimal(10,2)	YES		NULL	

```
5 rows in set (0.05 sec)
```

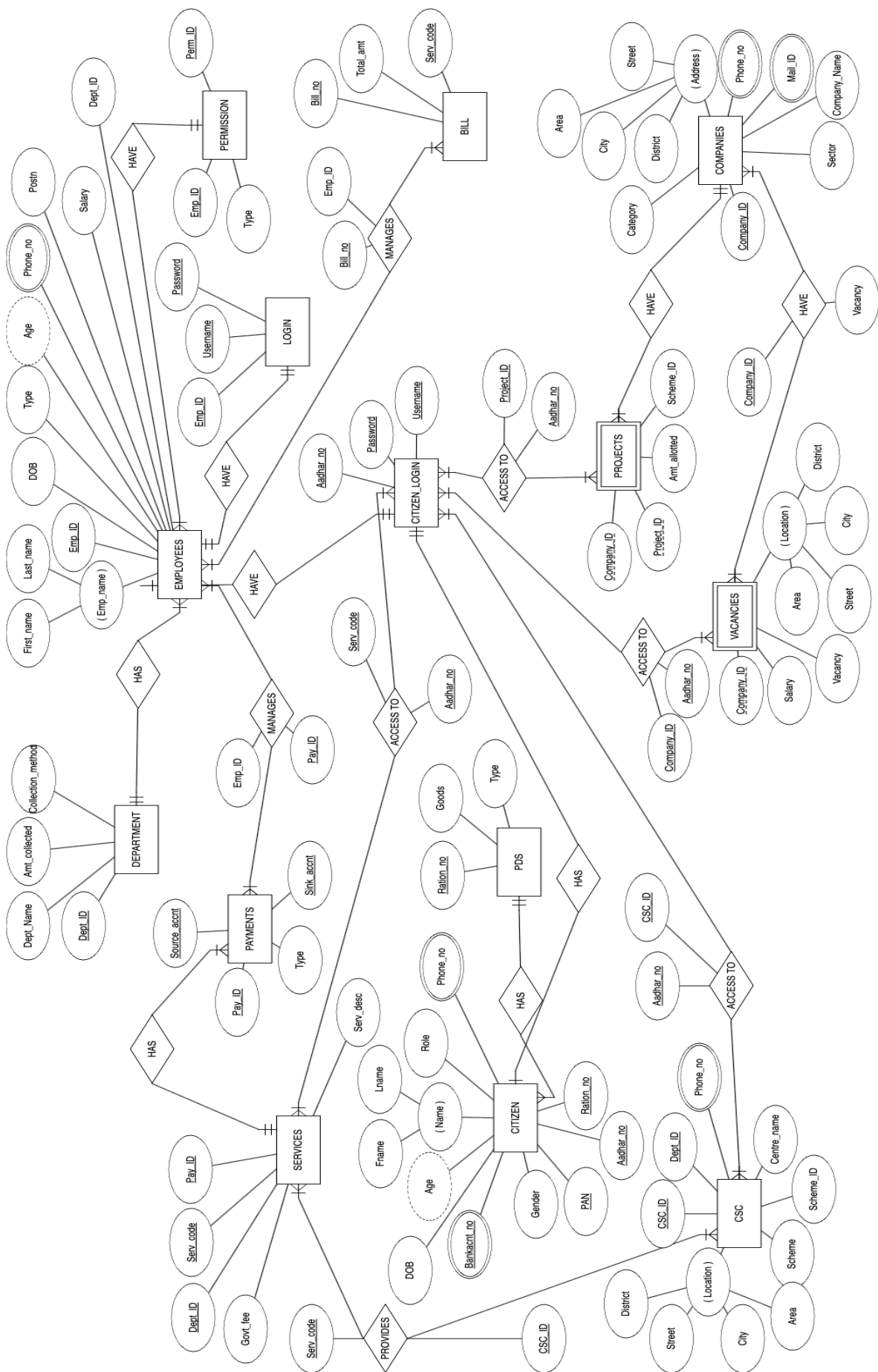
```
mysql> desc bill;
```

Field	Type	Null	Key	Default	Extra
Billno	int(11)	NO	PRI	NULL	
total	decimal(10,2)	YES		NULL	
servicecode	int(11)	YES	MUL	NULL	
emp_id	int(11)	YES	MUL	NULL	

```
4 rows in set (0.06 sec)
```

```
mysql> |
```

# ERD:



## IMPLEMENTATION:

### FLASK CONNECTIVITY

Code:

```
from flask import Flask, render_template, request, redirect, url_for, flash, session, jsonify
from flask_mysql import MySQL
from functools import wraps
import MySQLdb.cursors
import re
import hashlib

app = Flask(__name__)

# MySQL configurations
app.config['MYSQL_HOST'] = 'localhost'
app.config['MYSQL_USER'] = 'root'
app.config['MYSQL_PASSWORD'] = '1234'
app.config['MYSQL_DB'] = 'e_governance'
app.config['SECRET_KEY'] = 'your-secret-key'

mysql = MySQL(app)

# Login decorator for securing routes
def login_required(f):
    @wraps(f)
    def decorated_function(*args, **kwargs):
        if 'logged_in' not in session:
            return redirect(url_for('login'))
        return f(*args, **kwargs)
    return decorated_function

# Employee login decorator
```

```

def employee_required(f):
    @wraps(f)
    def decorated_function(*args, **kwargs):
        if 'user_type' not in session or session['user_type'] != 'employee':
            return redirect(url_for('login'))
        return f(*args, **kwargs)
    return decorated_function

@app.route('/')
def home():
    return render_template('index.html')

@app.route('/login', methods=['GET', 'POST'])
def login():
    if request.method == 'POST':
        user_type = request.form['user_type']
        username = request.form['username']
        password = request.form['password']

        cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)

        if user_type == 'employee':
            cursor.execute('SELECT * FROM Login WHERE Uname = %s AND pwd = %s', (username,
password,))
            account = cursor.fetchone()
            if account:
                session['logged_in'] = True
                session['user_type'] = 'employee'
                session['username'] = username
                session['emp_id'] = account['emp_id']
                return redirect(url_for('employee_dashboard'))
            else:
                cursor.execute('SELECT * FROM Citizen_Login WHERE Uname = %s AND pwd = %s',
(username, password,))

```

```

account = cursor.fetchone()

if account:

    session['logged_in'] = True

    session['user_type'] = 'citizen'

    session['username'] = username

    session['aadhar'] = account['Aadhar']

    return redirect(url_for('citizen_dashboard'))


flash('Invalid username/password!')

return render_template('login.html')


@app.route('/employee/dashboard')
@login_required
@employee_required
def employee_dashboard():

    cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)

    # Get employee permissions
    cursor.execute("""
        SELECT p.type
        FROM Permission p
        JOIN Employees e ON p.emp_id = e.emp_id
        WHERE e.emp_id = %s
    """, (session['emp_id'],))

    permissions = cursor.fetchall()

    return render_template('employee_dashboard.html', permissions=permissions)


@app.route('/citizen/dashboard')
@login_required
def citizen_dashboard():

    cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)

```

```

cursor.execute('SELECT * FROM Services')

services = cursor.fetchall()


# Get vacancies

cursor.execute("""
    SELECT c.name, v.locality, v.vacancies, v.salary
    FROM Vacancies v
    JOIN Companies c ON v.Company_id = c.Company_id
""")

vacancies = cursor.fetchall()


# Get PDS details (if needed)

cursor.execute('SELECT * FROM PDS WHERE Aadhar = %s', (session['aadhar'],))

pds_details = cursor.fetchall()


return render_template('citizen_dashboard.html',
    services=services,
    vacancies=vacancies,
    pds_details=pds_details,
    view="dashboard")


@app.route('/view_csc_centers')
@login_required
def view_csc_centers():
    cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)

    # Get CSC center details along with the services they provide
    cursor.execute("""
        SELECT c.csc_id, c.centre_name, c.location, c.Phone_no, s.serv
        FROM csc c
        JOIN service_csc sc ON c.csc_id = sc.csc_id
        JOIN services s ON sc.serv_code = s.serv_code
    """)

```

```

csc_services = cursor.fetchall()

# Organize the data to group services by each CSC center
csc_centers = {}

for row in csc_services:
    csc_id = row['csc_id']
    if csc_id not in csc_centers:
        csc_centers[csc_id] = {
            'centre_name': row['centre_name'],
            'location': row['location'],
            'Phone_no': row['Phone_no'],
            'services': []
        }
    csc_centers[csc_id]['services'].append(row['serv'])

# Convert dictionary to list for easier handling in template
csc_centers_list = list(csc_centers.values())

return render_template('citizen_dashboard.html',
                       csc_centers=csc_centers_list,
                       view="csc_centers")

def get_primary_key_column(table_name):
    """
    Retrieves the primary key column name for the given table.
    """
    cursor = mysql.connection.cursor()
    cursor.execute(f"SHOW KEYS FROM {table_name} WHERE Key_name = 'PRIMARY'")
    result = cursor.fetchone()
    cursor.close()
    return result[4] if result else None # Column name is in the fifth field (index 4)

```

```

@app.route('/employee/manage/<table_name>', methods=['GET', 'POST', 'PUT', 'DELETE'])
@login_required
@employee_required
def manage_table(table_name):
    cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)

    # Get the primary key column dynamically
    primary_key_column = get_primary_key_column(table_name)
    if not primary_key_column:
        flash(f"Primary key not found for table {table_name}.")
        return redirect(url_for('employee_dashboard'))

    # Fetch all records for display
    if request.method == 'GET':
        cursor.execute(f'SELECT * FROM {table_name}')
        data = cursor.fetchall()

        return render_template('manage_table.html', table_name=table_name, data=data,
                               primary_key_column=primary_key_column)

    # Handle INSERT operation
    elif request.method == 'POST':
        columns = ', '.join(request.form.keys())
        values = tuple(request.form.values())
        placeholders = ', '.join(['%s'] * len(values))

        cursor.execute(f'INSERT INTO {table_name} ({columns}) VALUES ({placeholders})', values)
        mysql.connection.commit()
        flash(f'Record added successfully to {table_name}')
        return redirect(url_for('manage_table', table_name=table_name))

    # Handle DELETE operation
    elif request.method == 'DELETE':
        record_key = request.args.get(primary_key_column)
        cursor.execute(f'DELETE FROM {table_name} WHERE {primary_key_column} = %s', (record_key,))

```



```

mysql.connection.commit()

return jsonify(success=True)

# Handle UPDATE operation
elif request.method == 'PUT':
    record_key = request.args.get(primary_key_column)
    updates = ', '.join([f'{k} = %s' for k in request.form.keys()])
    values = list(request.form.values()) + [record_key]

    cursor.execute(f'UPDATE {table_name} SET {updates} WHERE {primary_key_column} = %s',
values)

    mysql.connection.commit()
    flash(f'Record updated successfully in {table_name}')
    return jsonify(success=True)

cursor.close()

# Citizen specific routes
@app.route('/citizen/services')
@login_required
def view_services():
    cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
    cursor.execute('SELECT * FROM Services')
    services = cursor.fetchall()
    return render_template('services.html', services=services)

@app.route('/citizen/vacancies')
@login_required
def view_vacancies():
    cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
    cursor.execute("""
        SELECT c.name, v.locality, v.vacancies, v.salary
        FROM Vacancies v
        JOIN Companies c ON v.Company_id = c.Company_id
    """)

```

```

    ")

    vacancies = cursor.fetchall()

    return render_template('vacancies.html', vacancies=vacancies)

@app.route('/citizen/csc')
@login_required
def view_csc():

    cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)

    cursor.execute("""
        SELECT c.*, d.Dept_name
        FROM CSC c
        JOIN Department d ON c.Dept_id = d.Dept_id
    """)

    csc_centers = cursor.fetchall()

    return render_template('csc.html', csc_centers=csc_centers)

@app.route('/logout')
def logout():

    session.pop('logged_in', None)
    session.pop('user_type', None)
    session.pop('username', None)
    return redirect(url_for('home'))

if __name__ == '__main__':
    app.run(debug=True)

```

## HTML FILES:

### Index.html

```
<!DOCTYPE html>
<html>
<head>
  <title>E-Governance Portal</title>
  <link href="https://cdnjs.cloudflare.com/ajax/libs/tailwindcss/2.2.19/tailwind.min.css" rel="stylesheet">
</head>
<body class="bg-gray-100">
  <div class="container mx-auto px-4 py-8">
    <h1 class="text-4xl font-bold text-center mb-8">E-Governance Portal</h1>
    <div class="flex justify-center space-x-4">
      <a href="{{ url_for('login') }}" class="bg-blue-500 text-white px-6 py-2 rounded-lg">Login</a>
    </div>
  </div>
</body>
</html>
```

### Login.html

```
<!DOCTYPE html>
<html>
<head>
  <title>Login - E-Governance Portal</title>
  <link href="https://cdnjs.cloudflare.com/ajax/libs/tailwindcss/2.2.19/tailwind.min.css" rel="stylesheet">
</head>
<body class="bg-gray-100">
  <div class="container mx-auto px-4 py-8">
    <div class="max-w-md mx-auto bg-white rounded-lg shadow-md p-6">
      <h2 class="text-2xl font-bold mb-6 text-center">Login</h2>
      <form method="POST" action="{{ url_for('login') }}">
        <div class="mb-4">
          <label class="block text-gray-700 mb-2">User Type</label>
```

```

        <select name="user_type" class="w-full px-3 py-2 border rounded-lg">
            <option value="employee">Employee</option>
            <option value="citizen">Citizen</option>
        </select>
    </div>
    <div class="mb-4">
        <label class="block text-gray-700 mb-2">Username</label>
        <input type="text" name="username" class="w-full px-3 py-2 border rounded-lg">
    </div>
    <div class="mb-6">
        <label class="block text-gray-700 mb-2">Password</label>
        <input type="password" name="password" class="w-full px-3 py-2 border rounded-lg">
    </div>
    <button type="submit" class="w-full bg-blue-500 text-white py-2 rounded-lg">Login</button>
</form>
</div>
</div>
</body>
</html>

```

## Employee\_Dashboard.html

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>{% block title %}Employee Dashboard{% endblock %}</title>
    <!-- Include Tailwind CSS -->
    <script src="https://cdn.tailwindcss.com"></script>
</head>
<body class="bg-gray-100 min-h-screen">
    <!-- Navigation -->

```

```

<nav class="bg-white shadow-lg">
  <div class="space-y-6">
    <h1 class="text-2xl font-bold">Employee Dashboard</h1>

    <div class="bg-white p-6 rounded-lg shadow">
      <h2 class="text-xl font-bold mb-4">Your Permissions</h2>
      <div class="space-y-2">
        {% for permission in permissions %}
          <div class="p-2 bg-gray-100 rounded">
            {{ permission.type }}
          </div>
        {% endfor %}
      </div>
    </div>

    <div class="bg-white p-6 rounded-lg shadow">
      <h2 class="text-xl font-bold mb-4">Manage Data</h2>
      <div class="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 gap-4">
        <a href="{{ url_for('manage_table', table_name='Services') }}"
          class="p-4 bg-blue-100 rounded hover:bg-blue-200">
          Manage Services
        </a>
        <a href="{{ url_for('manage_table', table_name='Department') }}"
          class="p-4 bg-blue-100 rounded hover:bg-blue-200">
          Manage Departments
        </a>
        <a href="{{ url_for('manage_table', table_name='CSC') }}"
          class="p-4 bg-blue-100 rounded hover:bg-blue-200">
          Manage CSC Centers
        </a>
      </div>
    </div>
  </div>
</body>
</html>

```

## Citizen\_Dashboard.html

```
<!DOCTYPE html>

<html>

<head>

  <title>Citizen Dashboard - E-Governance Portal</title>

  <link href="https://cdnjs.cloudflare.com/ajax/libs/tailwindcss/2.2.19/tailwind.min.css" rel="stylesheet">

</head>

<body class="bg-gray-100">

  <nav class="bg-white shadow-lg mb-8">

    <div class="container mx-auto px-4">

      <div class="flex justify-between items-center py-4">

        <div class="text-xl font-bold">Citizen Dashboard</div>

        <div class="space-x-4">

          <a href="{{ url_for('citizen_dashboard') }}" class="text-blue-500">Dashboard</a>

          <a href="{{ url_for('view_csc_centers') }}" class="text-blue-500">CSC Centers</a>

          <a href="{{ url_for('logout') }}" class="text-red-500">Logout</a>

        </div>

      </div>

    </div>

  </nav>

  <div class="container mx-auto px-4">

    <div class="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 gap-6">

      {% if view == "dashboard" %}

        <!-- Services Section -->

        <div class="bg-white rounded-lg shadow-md p-6">

          <h3 class="text-xl font-bold mb-4">Available Services</h3>

          <ul class="space-y-2">

            {% for service in services %}

              <li class="p-2 bg-gray-50 rounded">

                {{ service.serv }} - ₹{{ service.govt_fee }}

              </li>

            {% endfor %}

          </ul>

        </div>

      </div>

    </div>

  </div>

</body>

</html>
```

```

</div>

<!-- Vacancies Section -->

<div class="bg-white rounded-lg shadow-md p-6">

  <h3 class="text-xl font-bold mb-4">Latest Vacancies</h3>

  <ul class="space-y-2">

    {% for vacancy in vacancies %}

      <li class="p-2 bg-gray-50 rounded">

        {{ vacancy.name }} - {{ vacancy.locality }}

        <div class="text-sm text-gray-600">

          Positions: {{ vacancy.vacancies }} | Salary: ₹{{ vacancy.salary }}

        </div>

      </li>

      {% endfor %}

    </ul>

  </div>

{% endif %}

{% if view == "csc_centers" %}

  <!-- CSC Center Services Section -->

  <div class="bg-white rounded-lg shadow-md p-6">

    <h3 class="text-xl font-bold mb-4">CSC Center Services</h3>

    <ul class="space-y-2">

      {% for center in csc_centers %}

        <li class="p-2 bg-gray-50 rounded">

          <strong>{{ center.centre_name }} ({{ center.location }})</strong><br>

          Phone: {{ center.Phone_no }}

          <ul class="mt-2 ml-4 list-disc">

            {% for service in center.services %}

              <li>{{ service }}</li>

            {% endfor %}

          </ul>

        </li>

      {% endfor %}

    </ul>

  </div>

{% endif %}


```

```

        {% endif %}

    </div>

</div>

</body></html>

```

## Manage\_table.html

```

<!DOCTYPE html>

<html>

<head>

    <title>Manage {{ table_name }} - E-Governance Portal</title>

    <link href="https://cdnjs.cloudflare.com/ajax/libs/tailwindcss/2.2.19/tailwind.min.css" rel="stylesheet">

</head>

<body class="bg-gray-100">

    <nav class="bg-white shadow-lg mb-8">

        <div class="container mx-auto px-4">

            <div class="flex justify-between items-center py-4">

                <div class="text-xl font-bold">Manage {{ table_name }}</div>

                <div class="space-x-4">

                    <a href="{{ url_for('employee_dashboard') }}" class="text-blue-500">Dashboard</a>

                    <a href="{{ url_for('logout') }}" class="text-red-500">Logout</a>

                </div>

            </div>

        </div>

    </nav>

    <div class="container mx-auto px-4">

        <!-- Add New Record Form -->

        <div class="bg-white rounded-lg shadow-md p-6 mb-6">

            <h3 class="text-xl font-bold mb-4">Add New Record</h3>

            <form id="addForm" class="grid grid-cols-2 gap-4" onsubmit="addRecord(event)">

                {% for column in data[0].keys() %}

                <div class="mb-4">

                    <label class="block text-gray-700 mb-2">{{ column }}</label>

                    <input type="text" name="{{ column }}" class="w-full px-3 py-2 border rounded-lg">

                </div>

            </form>

        </div>

    </div>

</body>

</html>

```



```

</div>

{%% endfor %}

<div class="col-span-2">

    <button type="submit" class="bg-green-500 text-white px-4 py-2 rounded-lg">Add
Record</button>

</div>

</form>

</div>

<div class="bg-white rounded-lg shadow-md p-6">

    <h3 class="text-xl font-bold mb-4">Existing Records</h3>

    <div class="overflow-x-auto">

        <table class="min-w-full table-auto">

            <thead>

                <tr>

                    {%% for column in data[0].keys() %}

                    <th class="px-4 py-2 bg-gray-50">{{ column }}</th>

                    {%% endfor %}

                    <th class="px-4 py-2 bg-gray-50">Actions</th>

                </tr>

            </thead>

            <tbody>

                {%% for row in data %}

                <tr class="border-t" data-key="{{ row[primary_key_column] }}">

                    {%% for value in row.values() %}

                    <td class="px-4 py-2">{{ value }}</td>

                    {%% endfor %}

                    <td class="px-4 py-2">

                        <button onclick="showEditModal('{{ row[primary_key_column] }}')" class="bg-blue-
500 text-white px-2 py-1 rounded mr-2">Edit</button>

                        <button onclick="deleteRecord('{{ row[primary_key_column] }}')" class="bg-red-500
text-white px-2 py-1 rounded">Delete</button>

                    </td>

                </tr>

                {%% endfor %}

            </tbody>

        </table>

    </div>

</div>


```

```

        </tbody>
    </table>
</div>
</div>
</div>

```

```

<div id="editModal" class="fixed inset-0 hidden bg-gray-800 bg-opacity-50 flex items-center justify-center">
    <div class="bg-white rounded-lg shadow-md p-6 w-full max-w-lg">
        <h3 class="text-xl font-bold mb-4">Edit Record</h3>
        <form id="editForm" class="grid grid-cols-2 gap-4" onsubmit="updateRecord(event)">
            {% for column in data[0].keys() %}
            <div class="mb-4">
                <label class="block text-gray-700 mb-2">{{ column }}</label>
                <input type="text" name="{{ column }}" id="edit-{{ column }}" class="w-full px-3 py-2 border rounded-lg">
            </div>
            {% endfor %}
            <div class="col-span-2 flex justify-end space-x-4">
                <button type="button" onclick="closeEditModal()" class="bg-gray-500 text-white px-4 py-2 rounded-lg">Cancel</button>
                <button type="submit" class="bg-blue-500 text-white px-4 py-2 rounded-lg">Save Changes</button>
            </div>
        </form>
    </div>
</div>

<script>
    let currentKey = null;

    function addRecord(event) {
        event.preventDefault();
        const formData = new FormData(document.getElementById("addForm"));
    }

```

```

fetch(`/employee/manage/{{ table_name }}`, {
  method: 'POST',
  body: formData,
})
.then(response => response.json())
.then(data => location.reload());
}

function showEditModal(key) {
  currentKey = key;
  const row = document.querySelector(`tr[data-key='${key}']`);
  const cells = row.querySelectorAll('td');

  {% for column in data[0].keys() %}
    document.getElementById("edit-{{ column }}").value = cells[{{ loop.index0 }}].innerText;
  {% endfor %}

  document.getElementById("editModal").classList.remove("hidden");
}

function closeEditModal() {
  document.getElementById("editModal").classList.add("hidden");
  currentKey = null;
}

function updateRecord(event) {
  event.preventDefault();
  const formData = new FormData(document.getElementById("editForm"));

  fetch(`/employee/manage/{{ table_name }}?{{ primary_key_column }}=${currentKey}`, {
    method: 'PUT',
    body: formData,
  })

```

```

        .then(response => response.json())
        .then(data => {
            location.reload();
        });
    }

function deleteRecord(key) {
    if (confirm('Are you sure you want to delete this record?')) {
        fetch(`/employee/manage/{{ table_name }}?{{ primary_key_column }}=${key}`, {
            method: 'DELETE',
        })
        .then(response => response.json())
        .then(data => location.reload());
    }
}
</script>
</body>
</html>

```

## Services.html

```

<!DOCTYPE html>
<html>
<head>
    <title>Available Services - E-Governance Portal</title>
    <link href="https://cdnjs.cloudflare.com/ajax/libs/tailwindcss/2.2.19/tailwind.min.css" rel="stylesheet">
</head>
<body class="bg-gray-100">
    <nav class="bg-white shadow-lg mb-8">
        <div class="container mx-auto px-4">
            <div class="flex justify-between items-center py-4">
                <div class="text-xl font-bold">Available Services</div>
                <div class="space-x-4">

```

```

        <a href="{{ url_for('citizen_dashboard') }}" class="text-blue-500">Dashboard</a>

        <a href="{{ url_for('logout') }}" class="text-red-500">Logout</a>

    </div>

</div>

</div>

</nav>

<div class="container mx-auto px-4">

    <div class="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 gap-6">

        {% for service in services %}

        <div class="bg-white rounded-lg shadow-md p-6">

            <h3 class="text-xl font-bold mb-2">{{ service.serv }}</h3>

            <p class="text-gray-600 mb-4">Department: {{ service.Dept_id }}</p>

            <p class="text-lg font-semibold">Fee: ₹{{ service.govt_fee }}</p>

            <button class="mt-4 bg-blue-500 text-white px-4 py-2 rounded-lg w-full">Apply Now</button>

        </div>

        {% endfor %}

    </div>

</div>

</body>

</html>

```

## Vacancies.html

```

<!-- templates/vacancies.html -->

<!DOCTYPE html>

<html>

<head>

    <title>Job Vacancies - E-Governance Portal</title>

    <link href="https://cdnjs.cloudflare.com/ajax/libs/tailwindcss/2.2.19/tailwind.min.css" rel="stylesheet">

</head>

<body class="bg-gray-100">

    <nav class="bg-white shadow-lg mb-8">

        <div class="container mx-auto px-4">

            <div class="flex justify-between items-center py-4">

```

```

<div class="text-xl font-bold">Job Vacancies</div>

<div class="space-x-4">

  <a href="{{ url_for('citizen_dashboard') }}" class="text-blue-500">Dashboard</a>

  <a href="{{ url_for('logout') }}" class="text-red-500">Logout</a>

</div>

</div>

</div>

</nav>

<div class="container mx-auto px-4">

  <div class="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 gap-6">

    {% for vacancy in vacancies %}

      <div class="bg-white rounded-lg shadow-md p-6">

        <h3 class="text-xl font-bold mb-2">{{ vacancy.name }}</h3>

        <p class="text-gray-600 mb-2">Location: {{ vacancy.locality }}</p>

        <p class="text-gray-600 mb-2">Positions: {{ vacancy.vacancies }}</p>

        <p class="text-lg font-semibold mb-4">Salary: ₹{{ vacancy.salary }}</p>

        <button class="bg-blue-500 text-white px-4 py-2 rounded-lg w-full">Apply Now</button>

      </div>

    {% endfor %}

  </div>

</div>

</body>

</html>

```

## CSC.html

```

<!DOCTYPE html>

<html>

<head>

  <title>CSC Centers - E-Governance Portal</title>

  <link href="https://cdnjs.cloudflare.com/ajax/libs/tailwindcss/2.2.19/tailwind.min.css" rel="stylesheet">

</head>

<body class="bg-gray-100">

```

```

<nav class="bg-white shadow-lg mb-8">
  <div class="container mx-auto px-4">
    <div class="flex justify-between items-center py-4">
      <div class="text-xl font-bold">CSC Centers</div>
      <div class="space-x-4">
        <a href="{{ url_for('citizen_dashboard') }}" class="text-blue-500">Dashboard</a>
        <a href="{{ url_for('logout') }}" class="text-red-500">Logout</a>
      </div>
    </div>
  </div>
</nav>

<div class="container mx-auto px-4">
  <div class="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 gap-6">
    {% for center in csc_centers %}
      <div class="bg-white rounded-lg shadow-md p-6">
        <h3 class="text-xl font-bold mb-2">{{ center.centre_name }}</h3>
        <p class="text-gray-600 mb-2">Department: {{ center.Dept_name }}</p>
        <p class="text-gray-600 mb-2">Location: {{ center.location }}</p>
        <p class="text-gray-600 mb-2">Phone: {{ center.Phone_no }}</p>
        <button class="mt-4 bg-blue-500 text-white px-4 py-2 rounded-lg w-full">View
Services</button>
      </div>
    {% endfor %}
  </div>
</div>
</body>
</html>

```

## SCREENSHOT:

### Employees:

The screenshot shows a web browser window with the title "Login - E-Governance Portal". The address bar displays "127.0.0.1:5000/login". The main content area features a "Login" form with the following fields:

- User Type:** A dropdown menu with "Employee" selected.
- Username:** A text input field containing "johndoe".
- Password:** A text input field with masked characters "\*\*\*\*\*".
- Login:** A blue button labeled "Login".

The screenshot shows a web browser window with the title "Employee Dashboard". The address bar displays "127.0.0.1:5000/employee/dashboard". The main content area features the "Employee Dashboard" with the following sections:

- Your Permissions:** A section containing a single item, "Admin".
- Manage Data:** A section containing three buttons: "Manage Services", "Manage Departments", and "Manage CSC Centers".



Manage Services - E-Governance

127.0.0.1:5000/employee/manage/Services

DashboardLogout

Manage Services

Add New Record

serv\_code

Dept\_id

serv

govt\_fee

Add Record

Existing Records

serv_code	Dept_id	serv	govt_fee	Actions
101	1	Tax Filing	100.00	<div>EditDelete</div>
102	2	Employee Benefits	50.00	<div>EditDelete</div>
103	3	Ad Campaign	250.00	<div>EditDelete</div>
104	4	Supply Chain Management	150.00	<div>EditDelete</div>
105	3	Rally Application	1200.00	<div>EditDelete</div>

Manage Services - E-Governance

127.0.0.1:5000/employee/manage/Services

DashboardLogout

Manage Services

Add New Record

serv\_code

Dept\_id

serv

govt\_fee

Add Record

Existing Records

serv_code	Dept_id	serv	govt_fee	Actions
101	1	Tax Filing	100.00	<div>EditDelete</div>
102	2	Employee Benefits	50.00	<div>EditDelete</div>
103	3	Ad Campaign	200.00	<div>EditDelete</div>
104	4	Supply Chain Management	150.00	<div>EditDelete</div>

Edit Record

serv\_code

101

Dept\_id

1

serv

Tax Filing

govt\_fee

100.00

Cancel

Save Changes

Manage Department - E-Governan

127.0.0.1:5000/employee/manage/Department

DashboardLogout

Add New Record

Dept\_id

Dept\_name

amt\_coll

coll\_method

Add Record

Existing Records

Dept_id	Dept_name	amt_coll	coll_method	Actions
1	Finance	1000000.00	Online	<div>EditDelete</div>
2	HR	500000.00	Offline	<div>EditDelete</div>
3	Marketing	200000.00	Online	<div>EditDelete</div>
4	Operations	750000.00	Offline	<div>EditDelete</div>

Manage CSC - E-Governance Port

127.0.0.1:5000/employee/manage/CSC

DashboardLogout

Add New Record

csc\_id

Phone\_no

Dept\_id

centre\_name

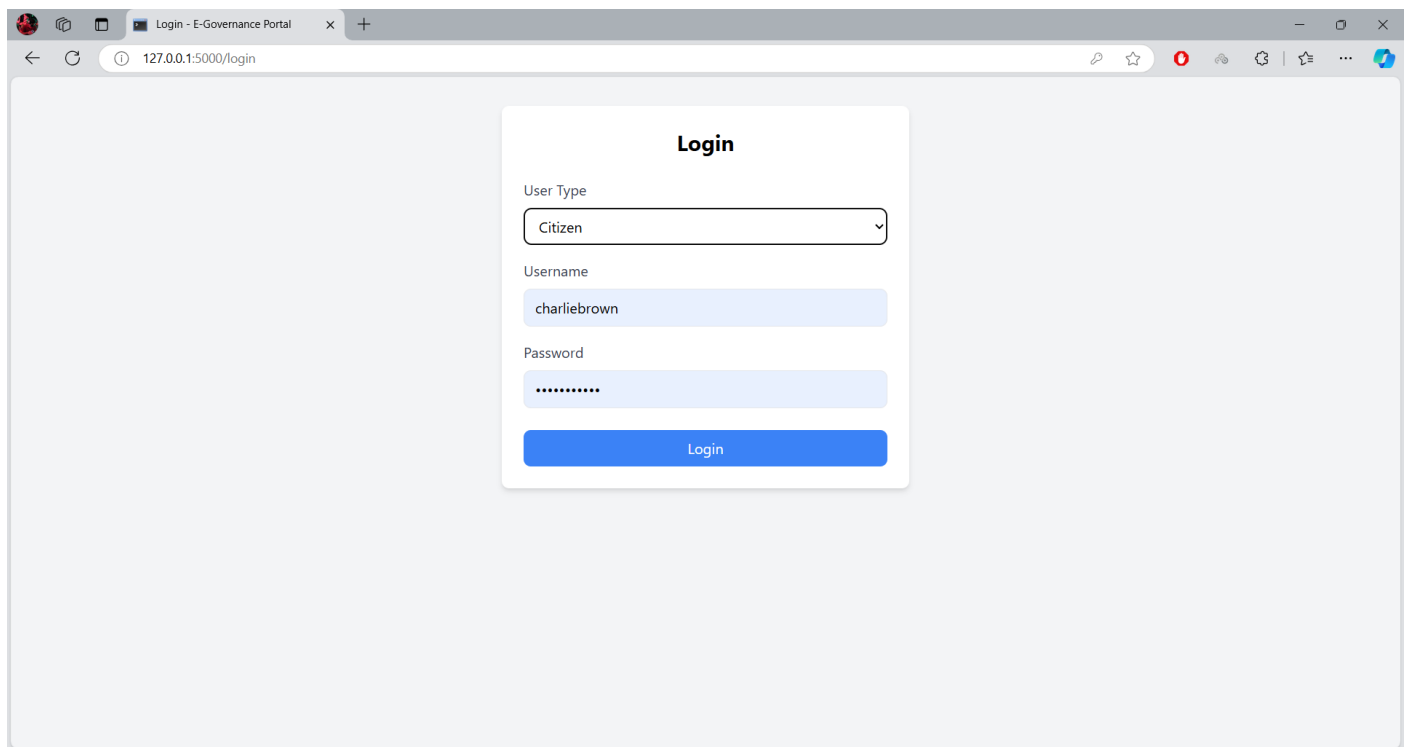
location

Add Record

Existing Records

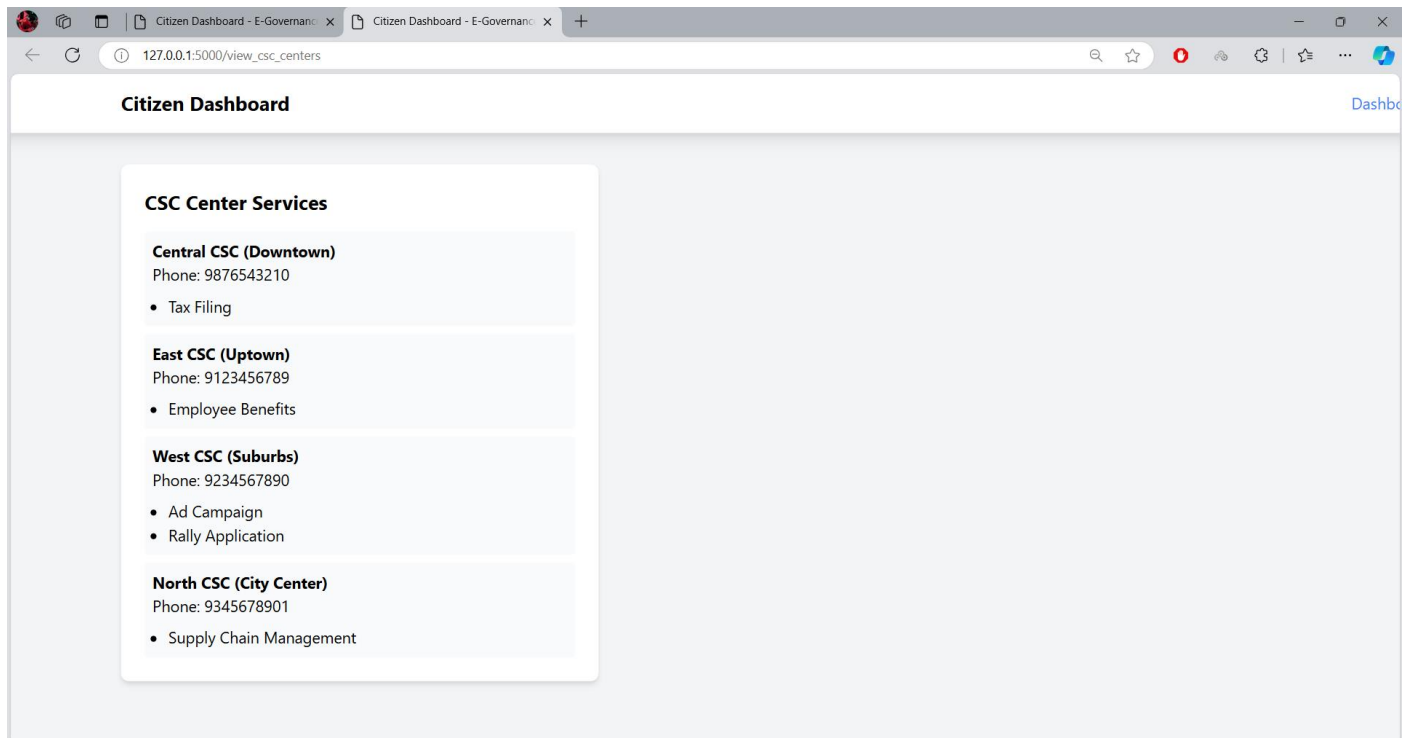
csc_id	Phone_no	Dept_id	centre_name	location	Actions
1	9876543210	1	Central CSC	Downtown	<div>EditDelete</div>
2	9123456789	2	East CSC	Uptown	<div>EditDelete</div>
3	9234567890	3	West CSC	Suburbs	<div>EditDelete</div>
4	9345678901	4	North CSC	City Center	<div>EditDelete</div>

# Citizen:



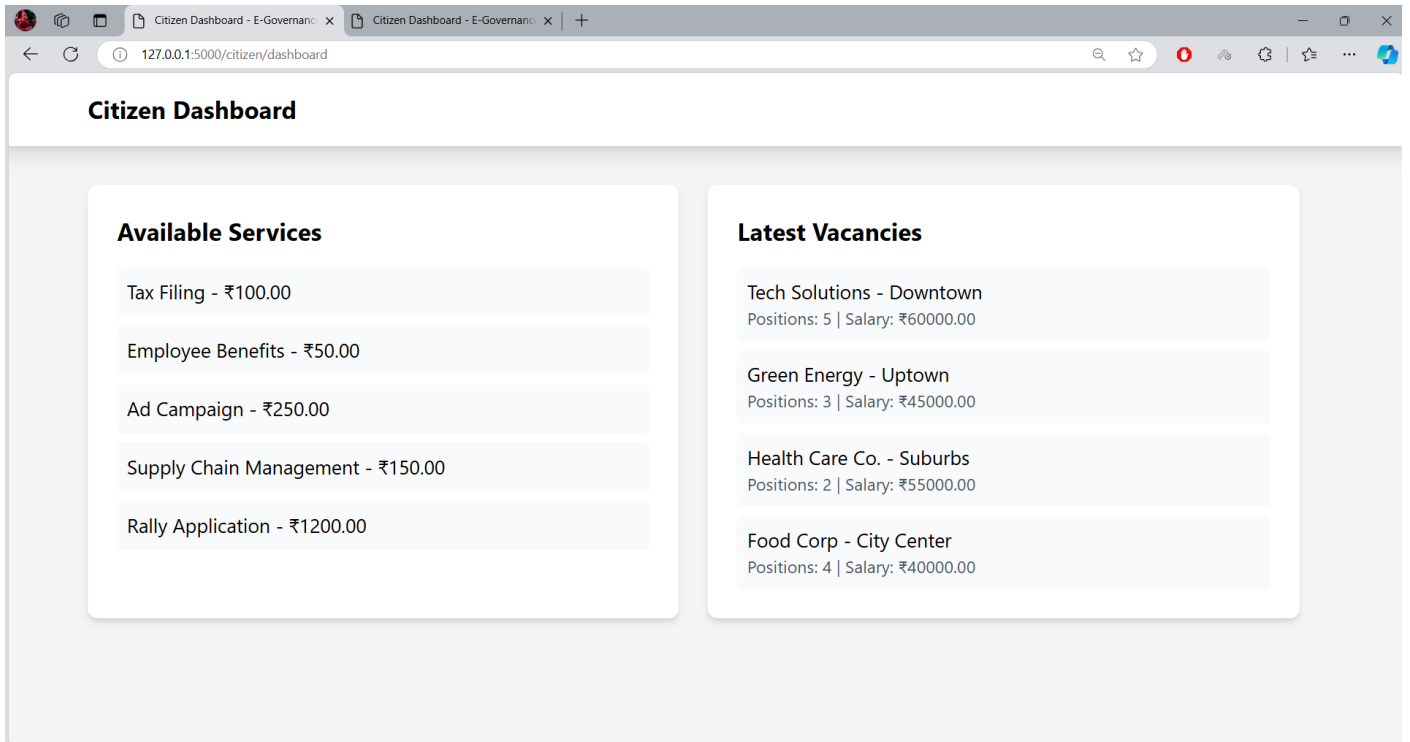
The screenshot shows a web browser window with the address bar displaying "127.0.0.1:5000/login". The page title is "Login - E-Governance Portal". The main content is a "Login" form with the following fields:

- User Type:** A dropdown menu with "Citizen" selected.
- Username:** A text input field containing "charliebrown".
- Password:** A password input field with masked characters "\*\*\*\*\*".
- Login:** A blue button labeled "Login".



The screenshot shows a web browser window with the address bar displaying "127.0.0.1:5000/view\_csc\_centers". The page title is "Citizen Dashboard - E-Governance Portal". The main content is a "Citizen Dashboard" with a sidebar menu on the left and a main content area on the right. The sidebar menu includes:

- CSC Center Services**
  - Central CSC (Downtown)**  
Phone: 9876543210
    - Tax Filing
  - East CSC (Uptown)**  
Phone: 9123456789
    - Employee Benefits
  - West CSC (Suburbs)**  
Phone: 9234567890
    - Ad Campaign
    - Rally Application
  - North CSC (City Center)**  
Phone: 9345678901
    - Supply Chain Management



## CONCLUSION:

The implemented system, built on MySQL, Python, Flask, HTML, and JavaScript, provides a robust and scalable platform for managing government services. Key strengths include data management, user authentication, citizen interfaces, and reporting. The modular architecture enables future enhancements like chatbots, mobile access, and data analytics. This implementation showcases the schema's effectiveness in delivering a comprehensive, adaptable, and citizen-focused solution.