Bank Management - Web Application

**A PROJECT REPORT**

*Submitted by*

**Sathyajit Suresh [Reg No: RA2211027010008]**

**C Anuchand [Reg No: RA2211027010015]**

**N Kushwant Sai [Reg No: RA2211027010018]**

**Hitesh Harish [Reg No: RA2211027010058]**

*Under the Guidance of*

**Dr. A.V. Kalpana**

(Assistant Professor, Department of Data Science and Business Systems)

*In partial fulfilment of the Requirements for the Degree*

*of*

**B.TECH COMPUTER SCIENCE WITH SPECIALISATION IN BIG DATA ANALYTICS**



**DEPARTMENT OF DATA SCIENCE AND BUSINESS SYSTEMS**

**FACULTY OF ENGINEERING AND TECHNOLOGY**

**SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**

INDEX

**SNO. TITLE PAGE NO.**

1 ABSTRACT 3

2 SOFTWARE REQUIREMENT 4

3 CODE 5

4 OUTPUT 24

1.ABSTRACT

The Bank Management System presented in this project embodies a sophisticated and efficient approach to modern banking operations, integrating Python, SQLite3, and Django to create a seamless and secure banking experience. In the contemporary financial landscape, where digitalization is paramount, this system serves as a pioneering solution, addressing the complexities of banking processes through innovative technology.

The system harnesses the flexibility and simplicity of Python, a versatile programming language, to deliver a user-friendly interface and robust functionality. Coupled with Django, a high-level web framework, the application ensures a responsive, scalable, and secure platform for both bank staff and customers. By adopting SQLite3 as the database management system, the project guarantees data integrity, reliability, and efficient storage and retrieval of critical banking information.

Key features of this Bank Management System include customer account management, transaction processing, loan management, and staff administration. The utilization of SQLite3 as the backend database ensures optimal performance and seamless data transactions, allowing the system to handle high volumes of concurrent users with ease.

This project represents a significant advancement in the digital transformation of banking institutions, offering enhanced customer experiences while optimizing internal banking operations. Through its modular architecture and real-time data processing capabilities, the system exemplifies the fusion of open-source technologies in revolutionizing the banking sector. Its adaptability and scalability make it an ideal choice for banks striving to meet the demands of a rapidly evolving digital landscape.

2.SOFTWARE REQUIREMENTS

**Python:** The project requires Python as the core programming language. Ensure you have Python installed on your system.

**Django Framework:** Django, a high-level Python web framework, is the backbone of this project. Install Django using pip:  
 pip install django

**SQLite Database:** SQLite is used as the built-in database management system. There is no need for separate installation, as SQLite comes bundled with Python. Ensure the Python SQLite library is available for database operations.

**Integrated Development Environment (IDE):** Choose a suitable IDE for Python development, such as PyCharm, Visual Studio Code, or Sublime Text.

**Database Management Tool:** For SQLite databases, you can use command-line tools or a database management tool like DB Browser for SQLite to interact with the database, create tables, and manage data.

**Version Control System (Optional):** Consider using version control tools like Git and platforms like GitHub for collaborative development and version management.

**Web Browser:** A modern web browser (e.g., Chrome, Firefox, Safari) for testing and debugging the web application's frontend and backend features.

**Operating System:** The project is platform-independent and can be developed and deployed on various operating systems, including Windows, macOS, and Linux distributions.

**Text Editor:** A basic text editor (e.g., Notepad, Sublime Text, Visual Studio Code) for editing configuration files, code snippets, and README files.

**Command Line Interface:** Familiarity with the command line or terminal for running Django management commands, installing packages, and navigating the project directory.

Ensure that you have these software components installed and configured to effectively develop, test, and deploy the Bank Management Web Application with Python and Django, utilising SQLite as the database system.

3.CODE

3.1. View:

from dateutil.relativedelta import relativedelta

from django.contrib import messages

from django.contrib.auth.mixins import LoginRequiredMixin

from django.urls import reverse\_lazy

from django.utils import timezone

from django.views.generic import CreateView, ListView

from transactions.constants import DEPOSIT, WITHDRAWAL

from transactions.forms import (

DepositForm,

TransactionDateRangeForm,

WithdrawForm,

)

from transactions.models import Transaction

class TransactionRepostView(LoginRequiredMixin, ListView):

template\_name = 'transactions/transaction\_report.html'

model = Transaction

form\_data = {}

def get(self, request, \*args, \*\*kwargs):

form = TransactionDateRangeForm(request.GET or None)

if form.is\_valid():

self.form\_data = form.cleaned\_data

return super().get(request, \*args, \*\*kwargs)

def get\_queryset(self):

queryset = super().get\_queryset().filter(

account=self.request.user.account

)

daterange = self.form\_data.get("daterange")

if daterange:

queryset = queryset.filter(timestamp\_date\_range=daterange)

return queryset.distinct()

def get\_context\_data(self, \*\*kwargs):

context = super().get\_context\_data(\*\*kwargs)

context.update({

'account': self.request.user.account,

'form': TransactionDateRangeForm(self.request.GET or None)

})

return context

class TransactionCreateMixin(LoginRequiredMixin, CreateView):

template\_name = 'transactions/transaction\_form.html'

model = Transaction

title = ''

success\_url = reverse\_lazy('transactions:transaction\_report')

def get\_form\_kwargs(self):

kwargs = super().get\_form\_kwargs()

kwargs.update({

'account': self.request.user.account

})

return kwargs

def get\_context\_data(self, \*\*kwargs):

context = super().get\_context\_data(\*\*kwargs)

context.update({

'title': self.title

})

return context

class DepositMoneyView(TransactionCreateMixin):

form\_class = DepositForm

title = 'Deposit Money to Your Account'

def get\_initial(self):

initial = {'transaction\_type': DEPOSIT}

return initial

def form\_valid(self, form):

amount = form.cleaned\_data.get('amount')

account = self.request.user.account

if not account.initial\_deposit\_date:

now = timezone.now()

next\_interest\_month = int(

12 / account.account\_type.interest\_calculation\_per\_year

)

account.initial\_deposit\_date = now

account.interest\_start\_date = (

now + relativedelta(

months=+next\_interest\_month

)

)

account.balance += amount

account.save(

update\_fields=[

'initial\_deposit\_date',

'balance',

'interest\_start\_date'

]

)

messages.success(

self.request,

f'{amount}$ was deposited to your account successfully'

)

return super().form\_valid(form)

class WithdrawMoneyView(TransactionCreateMixin):

form\_class = WithdrawForm

title = 'Withdraw Money from Your Account'

def get\_initial(self):

initial = {'transaction\_type': WITHDRAWAL}

return initial

def form\_valid(self, form):

amount = form.cleaned\_data.get('amount')

self.request.user.account.balance -= form.cleaned\_data.get('amount')

self.request.user.account.save(update\_fields=['balance'])

messages.success(

self.request,

f'Successfully withdrawn {amount}$ from your account'

)

return super().form\_valid(form)

3.2. Template:

3.2.1. User Login

{% extends 'core/base.html' %}

{% block head\_title %}Banking System{% endblock %}

{% block content %}

{% if form.non\_field\_errors %}

{% for error in form.non\_field\_errors %}

<div class="bg-red-100 border border-red-400 text-red-700 px-4 py-3 rounded relative mt-2" role="alert">

<p class="font-bold">Error!</p>

<p class="block sm:inline">{{ error }}</p>

</div>

{% endfor %}

{% endif %}

<h1 class="font-mono font-bold text-3xl text-center pb-5 pt-10">Sign In</h1>

<hr>

<div class="w-full mt-10">

<form method="post" class="bg-white shadow-md rounded px-8 pt-6 pb-8 mb-4">

{% csrf\_token %}

{% for hidden\_field in form.hidden\_fields %}

{{ hidden\_field.errors }}

{{ hidden\_field }}

{% endfor %}

{% for field in form.visible\_fields %}

<div class="mb-4">

<label class="block text-gray-700 text-sm font-bold mb-2" for="{{ field.id\_for\_label }}">

{{ field.label }}

</label>

<input class="shadow appearance-none border border-gray-500 rounded w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none focus:shadow-outline" name="{{ field.name }}" id="{{ field.id\_for\_label }}" type="{{ field.widget\_type }}">

</div>

{% if field.errors %}

{% for error in field.errors %}

<p class="text-red-600 text-sm italic pb-2">{{ error }}</p>

{% endfor %}

{% endif %}

{% endfor %}

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

<button class="bg-blue-500 hover:bg-blue-700 text-white font-bold py-2 px-4 rounded focus:outline-none focus:shadow-outline" type="submit">

Sign In

</button>

</div>

</form>

</div>

{% endblock %}

3.2.2. User Registration

{% extends 'core/base.html' %}

{% block head\_title %}Banking System{% endblock %}

{% block content %}

{% if registration\_form.non\_field\_errors %}

{% for error in registration\_form.non\_field\_errors %}

<div class="bg-red-100 border border-red-400 text-red-700 px-4 py-3 rounded relative mt-2" role="alert">

<p class="font-bold">Error!</p>

<p class="block sm:inline">{{ error }}</p>

</div>

{% endfor %}

{% endif %}

{% if address\_form.non\_field\_errors %}

{% for error in address\_form.non\_field\_errors %}

<div class="bg-red-100 border border-red-400 text-red-700 px-4 py-3 rounded relative mt-2" role="alert">

<p class="font-bold">Error!</p>

<p class="block sm:inline">{{ error }}</p>

</div>

{% endfor %}

{% endif %}

<h1 class="font-mono font-bold text-3xl text-center pb-5 pt-10">Register</h1>

<hr />

<div class="w-full mt-10">

<form method="post" class="bg-white shadow-md rounded px-8 pt-6 pb-8 mb-4">

{% csrf\_token %}

{% for hidden\_field in registration\_form.hidden\_fields %}

{{ hidden\_field.errors }}

{{ hidden\_field }}

{% endfor %}

<div class="flex flex-wrap -mx-3 mb-6">

<div class="w-full md:w-1/2 px-3 mb-6 md:mb-0">

<label class="block uppercase tracking-wide text-gray-700 text-xs font-bold mb-2" for="{{ registration\_form.first\_name.id\_for\_label }}">

{{ registration\_form.first\_name.label }}

</label>

{{ registration\_form.first\_name }}

{% if registration\_form.first\_name.errors %}

{% for error in registration\_form.first\_name.errors %}

<p class="text-red-600 text-sm italic pb-2">{{ error }}</p>

{% endfor %}

{% endif %}

</div>

<div class="w-full md:w-1/2 px-3">

<label class="block uppercase tracking-wide text-gray-700 text-xs font-bold mb-2" for="{{ registration\_form.last\_name.id\_for\_label }}">

{{ registration\_form.last\_name.label }}

</label>

{{ registration\_form.last\_name }}

{% if registration\_form.last\_name.errors %}

{% for error in registration\_form.last\_name.errors %}

<p class="text-red-600 text-sm italic pb-2">{{ error }}</p>

{% endfor %}

{% endif %}

</div>

</div>

<div class="flex flex-wrap -mx-3 mb-6">

<div class="w-full md:w-1/2 px-3 mb-6 md:mb-0">

<label class="block uppercase tracking-wide text-gray-700 text-xs font-bold mb-2" for="{{ registration\_form.email.id\_for\_label }}">

{{ registration\_form.email.label }}

</label>

{{ registration\_form.email }}

{% if registration\_form.email.errors %}

{% for error in registration\_form.email.errors %}

<p class="text-red-600 text-sm italic pb-2">{{ error }}</p>

{% endfor %}

{% endif %}

</div>

<div class="w-full md:w-1/2 px-3 mb-6 md:mb-0">

<label class="block uppercase tracking-wide text-gray-700 text-xs font-bold mb-2" for="{{ registration\_form.account\_type.id\_for\_label }}">

{{ registration\_form.account\_type.label }}

</label>

{{ registration\_form.account\_type }}

{% if registration\_form.account\_type.errors %}

{% for error in registration\_form.account\_type.errors %}

<p class="text-red-600 text-sm italic pb-2">{{ error }}</p>

{% endfor %}

{% endif %}

</div>

</div>

<div class="flex flex-wrap -mx-3 mb-6">

<div class="w-full md:w-1/2 px-3 mb-6 md:mb-0">

<label class="block uppercase tracking-wide text-gray-700 text-xs font-bold mb-2" for="{{ registration\_form.gender.id\_for\_label }}">

{{ registration\_form.gender.label }}

</label>

{{ registration\_form.gender }}

{% if registration\_form.gender.errors %}

{% for error in registration\_form.gender.errors %}

<p class="text-red-600 text-sm italic pb-2">{{ error }}</p>

{% endfor %}

{% endif %}

</div>

<div class="w-full md:w-1/2 px-3 mb-6 md:mb-0">

<label class="block uppercase tracking-wide text-gray-700 text-xs font-bold mb-2" for="{{ registration\_form.birth\_date.id\_for\_label }}">

{{ registration\_form.birth\_date.label }}

</label>

{{ registration\_form.birth\_date }}

{% if registration\_form.birth\_date.errors %}

{% for error in registration\_form.birth\_date.errors %}

<p class="text-red-600 text-sm italic pb-2">{{ error }}</p>

{% endfor %}

{% endif %}

</div>

</div>

<div class="flex flex-wrap -mx-3 mb-6">

<div class="w-full md:w-1/2 px-3 mb-6 md:mb-0">

<label class="block uppercase tracking-wide text-gray-700 text-xs font-bold mb-2" for="{{ registration\_form.password1.id\_for\_label }}">

{{ registration\_form.password1.label }}

</label>

{{ registration\_form.password1 }}

{% if registration\_form.password1.errors %}

{% for error in registration\_form.password1.errors %}

<p class="text-red-600 text-sm italic pb-2">{{ error }}</p>

{% endfor %}

{% endif %}

</div>

<div class="w-full md:w-1/2 px-3 mb-6 md:mb-0">

<label class="block uppercase tracking-wide text-gray-700 text-xs font-bold mb-2" for="{{ registration\_form.password2.id\_for\_label }}">

{{ registration\_form.password2.label }}

</label>

{{ registration\_form.password2 }}

{% if registration\_form.password2.errors %}

{% for error in registration\_form.password2.errors %}

<p class="text-red-600 text-sm italic pb-2">{{ error }}</p>

{% endfor %}

{% endif %}

</div>

</div>

{% for hidden\_field in address\_form.hidden\_fields %} {{ hidden\_field.errors }} {{ hidden\_field }} {% endfor %}

<div class="flex flex-wrap -mx-3 mb-6">

<div class="w-full md:w-1/2 px-3 mb-6 md:mb-0">

<label class="block uppercase tracking-wide text-gray-700 text-xs font-bold mb-2" for="{{ address\_form.street\_address.id\_for\_label }}">

{{ address\_form.street\_address.label }}

</label>

{{ address\_form.street\_address }}

{% if address\_form.street\_address.errors %}

{% for error in address\_form.street\_address.errors %}

<p class="text-red-600 text-sm italic pb-2">{{ error }}</p>

{% endfor %}

{% endif %}

</div>

<div class="w-full md:w-1/2 px-3">

<label class="block uppercase tracking-wide text-gray-700 text-xs font-bold mb-2" for="{{ address\_form.city.id\_for\_label }}">

{{ address\_form.city.label }}

</label>

{{ address\_form.city }}

{% if address\_form.city.errors %}

{% for error in address\_form.city.errors %}

<p class="text-red-600 text-sm italic pb-2">{{ error }}</p>

{% endfor %}

{% endif %}

</div>

</div>

<div class="flex flex-wrap -mx-3 mb-6">

<div class="w-full md:w-1/2 px-3 mb-6 md:mb-0">

<label class="block uppercase tracking-wide text-gray-700 text-xs font-bold mb-2" for="{{ address\_form.postal\_code.id\_for\_label }}">

{{ address\_form.postal\_code.label }}

</label>

{{ address\_form.postal\_code }}

{% if address\_form.postal\_code.errors %}

{% for error in address\_form.postal\_code.errors %}

<p class="text-red-600 text-sm italic pb-2">{{ error }}</p>

{% endfor %}

{% endif %}

</div>

<div class="w-full md:w-1/2 px-3">

<label class="block uppercase tracking-wide text-gray-700 text-xs font-bold mb-2" for="{{ address\_form.country.id\_for\_label }}">

{{ address\_form.country.label }}

</label>

{{ address\_form.country }}

{% if address\_form.country.errors %}

{% for error in address\_form.country.errors %}

<p class="text-red-600 text-sm italic pb-2">{{ error }}</p>

{% endfor %}

{% endif %}

</div>

</div>

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

<button class="bg-blue-500 hover:bg-blue-700 text-white font-bold py-2 px-4 rounded focus:outline-none focus:shadow-outline" type="submit">

Register

</button>

</div>

</form>

</div>

{% endblock %}

3.2.4. Transaction Form

{% extends 'core/base.html' %}

{% block head\_title %}{{ title }}{% endblock %}

{% block content %}

<h1 class="font-mono font-bold text-3xl text-center pb-5 pt-10">{{ title }}</h1>

<hr />

<div class="w-full mt-5">

<form method="post" class="bg-white shadow-md rounded px-8 pt-6 pb-8 mb-4">

{% csrf\_token %}

<div class="mb-4">

<label class="block text-gray-700 text-sm font-bold mb-2" for="amount">

Amount

</label>

<input class="shadow appearance-none border rounded w-full py-2 px-3 text-gray-700 leading-tight border rounded-md border-gray-500 focus:outline-none focus:shadow-outline" name="amount" id="amount" type="number" required placeholder="Amount">

</div>

{% if form.amount.errors %}

{% for error in form.amount.errors %}

<p class="text-red-600 text-sm italic pb-2">{{ error }}</p>

{% endfor %}

{% endif %}

<button class="bg-blue-500 hover:bg-blue-700 text-white font-bold py-2 px-4 rounded focus:outline-none focus:shadow-outline" type="submit">

Submit

</button>

</form>

</div>

{% endblock %}

3.2.5. Transaction Report

{% extends 'core/base.html' %}

{% block head\_title %}Transaction Report{% endblock %}

{% block head\_extra %}

<script type="text/javascript" src="https://cdn.jsdelivr.net/jquery/latest/jquery.min.js"></script>

<script type="text/javascript" src="https://cdn.jsdelivr.net/momentjs/latest/moment.min.js"></script>

<script type="text/javascript" src="https://cdn.jsdelivr.net/npm/daterangepicker/daterangepicker.min.js"></script>

<link rel="stylesheet" type="text/css" href="https://cdn.jsdelivr.net/npm/daterangepicker/daterangepicker.css" />

{% endblock %}

{% block content %}

<h1 class="font-mono font-bold text-3xl text-center pb-5 pt-10">Transaction Report</h1>

<hr />

<form method="get">

<div class="mt-10 h-10 pl-3 pr-2 bg-white border rounded-md border-gray-500 flex justify-between items-center relative">

<input type="search" name="daterange" id="search" placeholder="Filter using date range" value="{{ request.GET.daterange }}" class="appearance-none w-full outline-none focus:outline-none active:outline-none" />

<button type="submit" class="ml-1 outline-none focus:outline-none active:outline-none">

<svg fill="none" stroke="currentColor" stroke-linecap="round" stroke-linejoin="round" stroke-width="2" viewBox="0 0 24 24" class="w-6 h-6">

<path d="M21 21l-6-6m2-5a7 7 0 11-14 0 7 7 0 0114 0z"></path>

</svg>

</button>

</div>

{% if form.daterange.errors %}

{% for error in form.daterange.errors %}

<p class="text-red-600 text-sm italic">{{ error }}</p>

{% endfor %}

{% endif %}

</form>

<table class="table-auto mx-auto w-full mt-8">

<thead class="bg-gray-800 text-white">

<tr class="uppercase font-semibold text-sm">

<th class="px-4 py-2">Transaction Type</th>

<th class="px-4 py-2">Date</th>

<th class="px-4 py-2">Amount</th>

<th class="px-4 py-2">Balance After Transaction</th>

</tr>

</thead>

<tbody>

{% for transaction in object\_list %}

<tr class="{% cycle 'bg-gray-300' 'bg-white-100' %}">

<td class="border px-4 py-2 text-center">{{ transaction.get\_transaction\_type\_display }}</td>

<td class="border px-4 py-2 text-center">{{ transaction.timestamp }}</td>

<td class="border px-4 py-2 text-center">₹ {{ transaction.amount }}</td>

<td class="border px-4 py-2 text-center">₹ {{ transaction.balance\_after\_transaction }}</td>

</tr>

{% endfor %}

<tr class="bg-gray-600 text-white">

<th class="px-4 py-2 text-right" colspan="3">Final Balance</th>

<th class="px-4 py-2 text-center">₹ {{ account.balance }}</th>

</tr>

</tbody>

</table>

{% endblock %}

{% block footer\_extra %}

<script type="text/javascript">

$(function () {

$('input[name="daterange"]').daterangepicker({

autoUpdateInput: false,

locale: {

cancelLabel: "Clear",

},

});

$('input[name="daterange"]').on("apply.daterangepicker", function (ev, picker) {

$(this).val(picker.startDate.format("YYYY-MM-DD") + " - " + picker.endDate.format("YYYY-MM-DD"));

});

$('input[name="daterange"]').on("cancel.daterangepicker", function (ev, picker) {

$(this).val("");

});

});

</script>

{% endblock %}

3.2.6. Index

{% extends 'core/base.html' %}

{% block head\_title %}Banking System{% endblock %}

{% block content %}

<div class="container mx-auto flex flex-col md:flex-row items-center my-12 md:my-24">

<!--Left Col-->

<div class="flex flex-col w-full lg:w-1/2 justify-center items-start pt-12 pb-24 px-6">

<h1 class="font-bold text-4xl my-4">Banking System</h1>

<p class="leading-normal mb-4">

This is an Example web application created using django.

</p>

<div>

<a href="{% url 'accounts:user\_registration' %}" class="bg-transparent hover:bg-gray-800 text-gray-800 hover:text-white rounded shadow hover:shadow-lg py-2 px-4 border border-gray-900 hover:border-transparent">Register</a>

</div>

</div>

<!--Right Col-->

<div class="w-full lg:w-1/2 lg:py-6 text-center">

<h1 style="font-size: 12rem" class="font-black text-gray-800">₹₹₹</h1>

</div>

</div>

{% endblock %}

3.2.7. Messages

{% for message in messages %}

<div class="bg-teal-100 border-t-4 border-teal-500 rounded-b text-teal-900 px-4 py-3 shadow-md mt-2" role="alert">

<div class="flex">

<div class="py-1"><svg class="fill-current h-6 w-6 text-teal-500 mr-4" xmlns="http://www.w3.org/2000/svg" viewBox="0 0 20 20"><path d="M2.93 17.07A10 10 0 1 1 17.07 2.93 10 10 0 0 1 2.93 17.07zm12.73-1.41A8 8 0 1 0 4.34 4.34a8 8 0 0 0 11.32 11.32zM9 11V9h2v6H9v-4zm0-6h2v2H9V5z"/></svg></div>

<div>

<p class="font-bold">Success!</p>

<p class="text-sm">{{ message }}</p>

</div>

</div>

</div>

{% endfor %}

3.2.8. Navigation Bar

<nav class="flex items-center justify-between flex-wrap bg-gray-800 p-6">

<div class="flex items-center flex-shrink-0 text-white mr-6">

<span class="font-semibold text-xl tracking-tight">Banking System</span>

</div>

<div class="block lg:hidden">

<button class="flex items-center px-3 py-2 border rounded text-white border-white-400 hover:text-white hover:border-white">

<svg class="fill-current h-3 w-3" viewBox="0 0 20 20" xmlns="http://www.w3.org/2000/svg"><title>Menu</title><path d="M0 3h20v2H0V3zm0 6h20v2H0V9zm0 6h20v2H0v-2z"/></svg>

</button>

</div>

<div class="w-full block flex-grow lg:flex lg:items-center lg:w-auto">

{% if request.user.is\_authenticated %}

<div class="text-md lg:flex-grow">

<a href="{% url 'transactions:transaction\_report' %}" class="block mt-4 lg:inline-block lg:mt-0 text-white hover:text-white mr-4">

Transaction Report

</a>

<a href="{% url 'transactions:deposit\_money' %}" class="block mt-4 lg:inline-block lg:mt-0 text-white hover:text-white mr-4">

Deposit

</a>

<a href="{% url 'transactions:withdraw\_money' %}" class="block mt-4 lg:inline-block lg:mt-0 text-white hover:text-white">

Withdraw

</a>

</div>

<div>

<a href="{% url 'accounts:user\_logout' %}" class="inline-block font-medium text-sm px-4 py-2 leading-none border rounded text-white border-white hover:border-transparent hover:text-gray-800 hover:bg-white mt-4 lg:mt-0">Logout</a>

</div>

{% else %}

<div class="text-md lg:flex-grow"></div>

<div>

<a href="{% url 'accounts:user\_login' %}" class="mr-2 inline-block font-medium text-sm px-4 py-2 leading-none border rounded text-white border-white hover:border-transparent hover:text-gray-800 hover:bg-white mt-4 lg:mt-0">Login</a>

</div>

<div>

<a href="{% url 'accounts:user\_registration' %}" class="inline-block font-medium text-sm px-4 py-2 leading-none border rounded text-white border-white hover:border-transparent hover:text-gray-800 hover:bg-white mt-4 lg:mt-0">Register</a>

</div>

{% endif %}

</div>

</nav>

3.2.9. Base

{% load static %}

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8" />

<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no" />

<meta name="description" content="Dummy Banking System" />

<meta name="author" content="Maksudul Haque" />

<title>

{% block head\_title %}Banking System{% endblock %}

</title>

<link href="https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css" rel="stylesheet">

{% block head\_extra %}{% endblock %}

</head>

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

{% block body %}

{% include 'core/navbar.html' %}

<div class="lg:container lg:mx-auto">

{% include 'core/messages.html' %}

{% block content %}

<!-- Add content here -->

{% endblock %}

</div>

{% include 'core/footer.html' %}

{% endblock %}

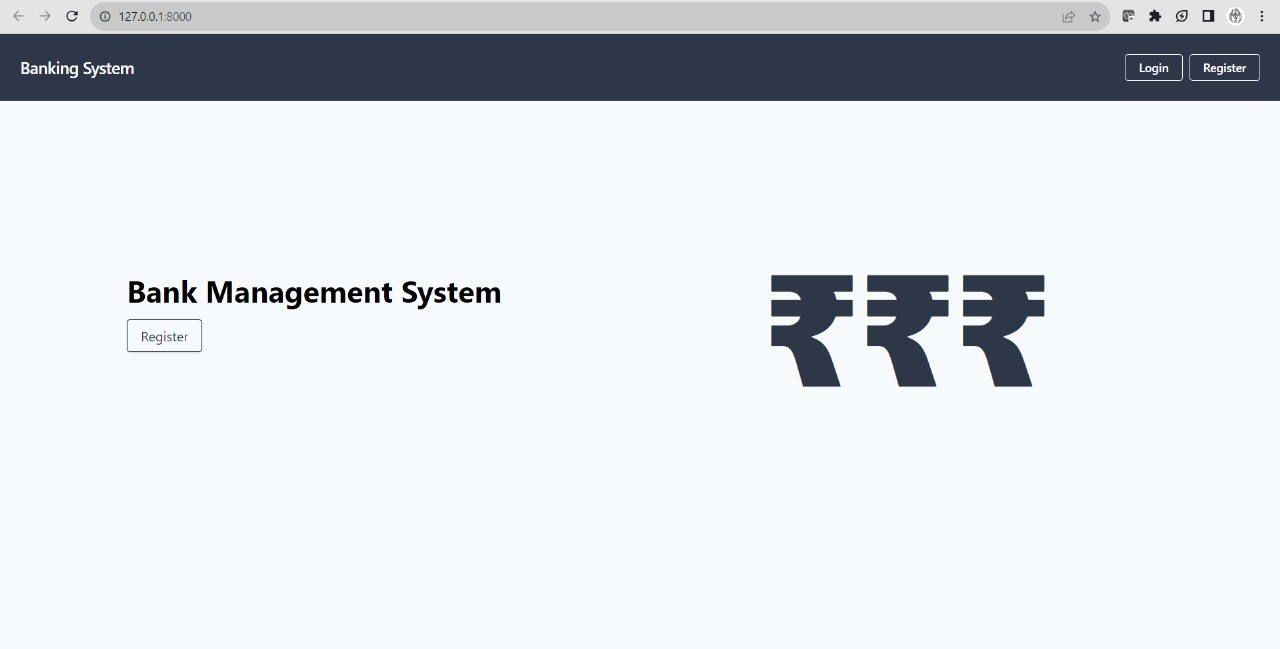
{% block footer\_extra %}{% endblock %}

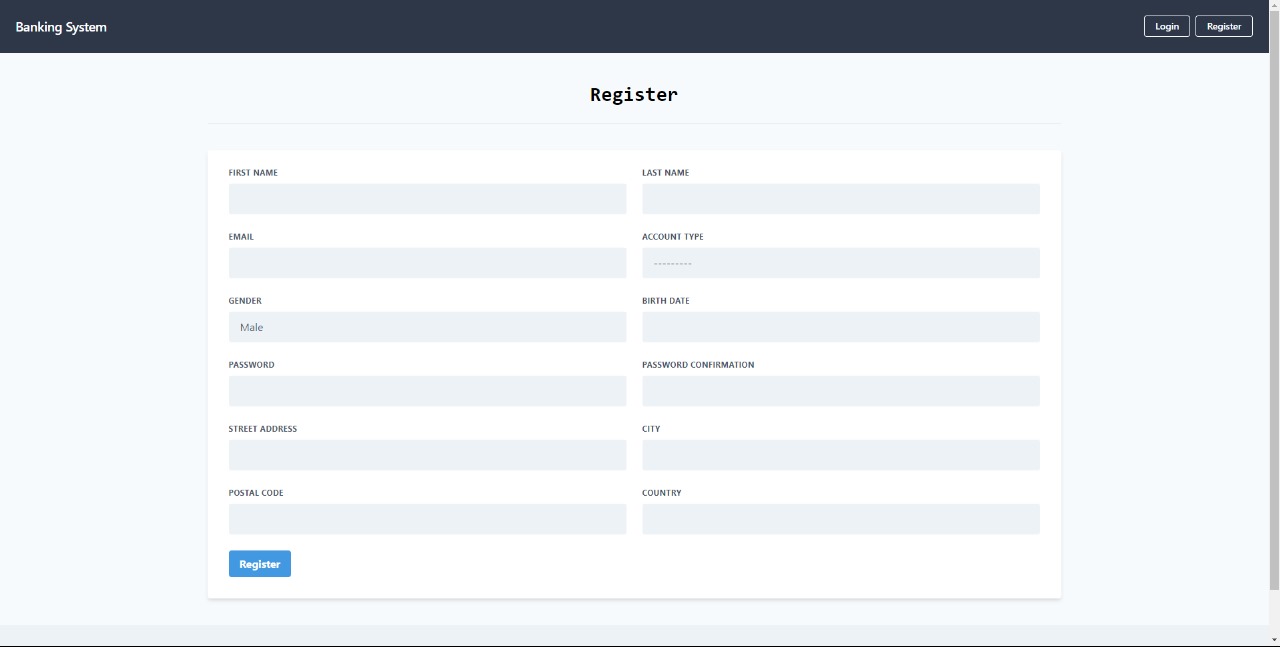
</body>

</html>

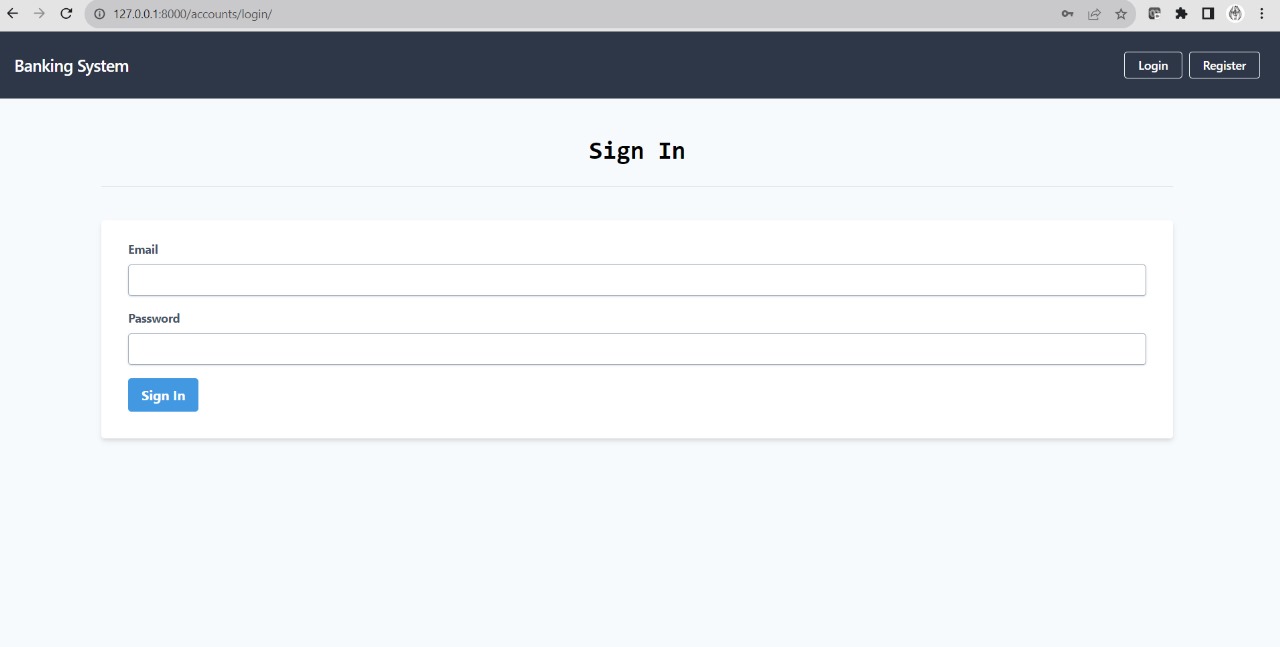
4.OUTPUT

Home

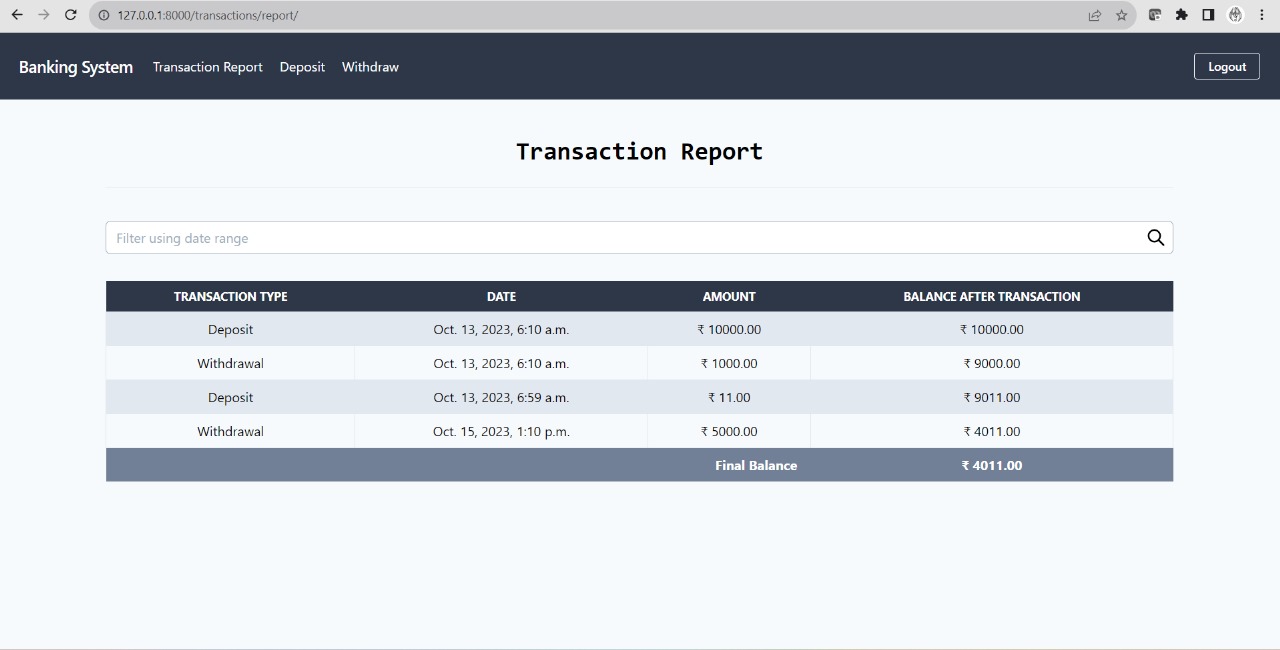


Registration

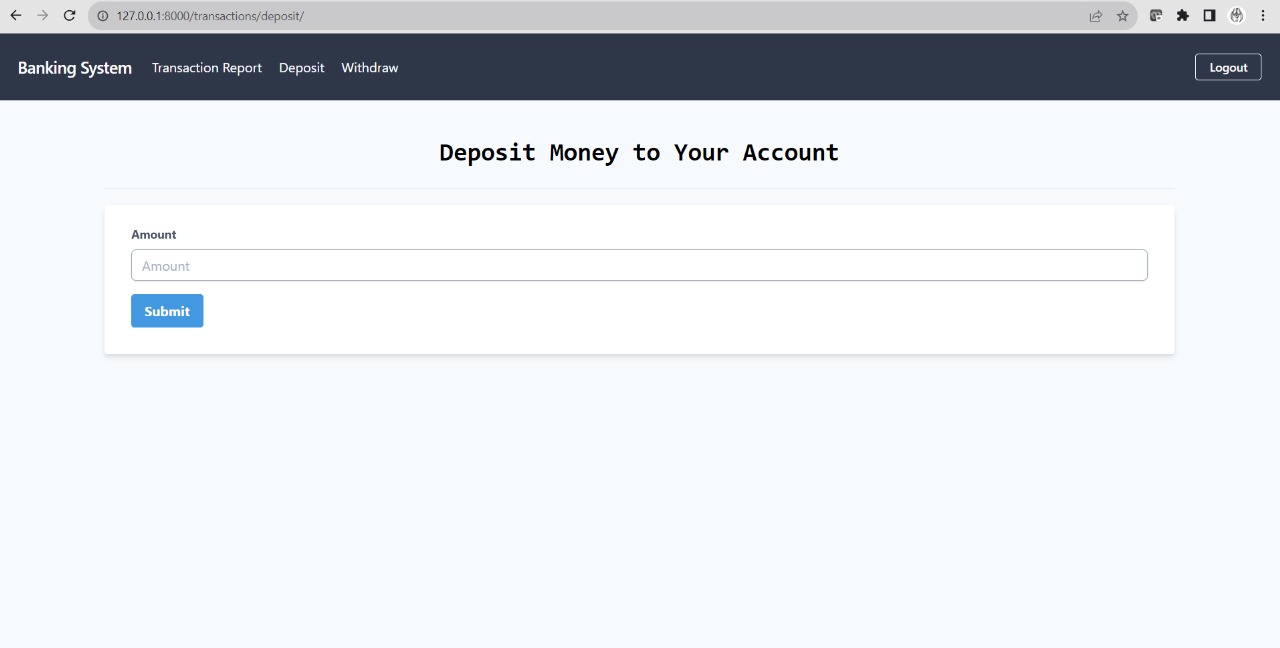
Sign In



Transaction Report



Deposit Money to Your Account



Withdraw Money from Your Account

