Visvesvaraya Technological University

BELAGAVI, KARNATAKA

ವಿಶ್ವೇಶ್ವರಯ್ಯ ತಾಂತ್ರಿಕ ವಿಶ್ವವಿದ್ಯಾಲಯ

ಬೆಳಗಾವಿ, ಕರ್ನಾಟಕ

Full Stack Development Project report on

"Small Finance Bank Account Management System"

Submitted by

1. Viresha H T	**4JN22IS183**
2. Yuvaraj M R	**4JN22IS187**

Under the guidance of

Mr. Arun Kumar B.E, MTech

Associate Professor, Dept. of IS&E,

JNNCE, Shivamogga.

Department of Information Science & Engineering
JNN College of Engineering
Shivamogga-577204

2024-25

TABLE OF CONTENTS

1. Introduction

Banking systems play a vital role in managing the financial transactions of individuals and businesses. With the rise of digital technology, there is a growing need for simple and accessible online banking solutions. This project aims to develop a basic Bank Account Management System using React.js. It allows account creation, secure login, and transaction management through a user-friendly web interface.

1.1 Overview of the Banking System

The banking system plays a crucial role in the financial stability of individuals and businesses by offering essential services such as savings, deposits, withdrawals, and money transfers. It acts as an intermediary between depositors and borrowers, ensuring a smooth flow of money in the economy. In recent years, banks have increasingly adopted digital platforms to improve accessibility and efficiency. With the rise of internet usage, online banking systems have become vital for real-time operations. These systems reduce the need for physical branch visits and enable users to perform transactions securely from anywhere. This project simulates a basic banking system using modern web technologies. It focuses on managing customer accounts, transactions, and balances efficiently. It provides the foundation for understanding digital transformation in banking services.

1.2 Motivation (Why this System is Useful in Real Life)

In the real world, customers expect convenience, speed, and security when managing their finances. Traditional banking methods often involve long queues and manual paperwork, which can be time-consuming and error-prone. A web-based system allows users to perform transactions, access their account details, and manage funds without visiting a branch. For bank staff, automating user registration and data management helps reduce operational overhead and minimizes human error. With the increasing demand for digital banking services, this project provides a foundational understanding of how modern banking platforms are built. It also serves as a learning model for aspiring developers interested in fintech. The simplicity of the system makes it easy to enhance with backend integration, authentication, or analytics in the future.

2. Problem Statement

Traditional banking systems often involve manual processes that are time-consuming and inefficient. Many small-scale or educational systems lack a user-friendly digital platform for managing customer accounts. Users face difficulty in accessing basic banking features like viewing balance or performing transactions online. There is a growing need for a simple, secure, and interactive system for account management. This project addresses these issues by building a web-based solution using React.js for efficient bank operations.

3. Objective of the Project

- To develop a simple and interactive web-based Bank Account Management System using React.js that allows smooth user interaction and navigation.
- To enable a bank manager to register new users, while allowing customers to securely log in using their account number and password.
- To provide basic banking functionalities such as checking account balance, and performing credit and debit operations.
- To help students understand frontend development concepts like component-based design, state management, routing, and local data storage using React.

4. Tools & Technologies Used

1. Frontend: React.js (JavaScript):

React.js is a popular JavaScript library for building user interfaces using reusable components. It helps in creating a dynamic, single-page web application efficiently. Its virtual DOM improves performance and responsiveness of the UI. React is ideal for developing modern web apps with interactive features.

2. Routing: React Router DOM

React Router DOM is used to handle navigation between different pages in a React app.It allows the app to switch views without reloading the entire page. Routes help define user paths like registration, login, and dashboard.This enables a smooth user experience in single-page applications

3. Data Storage: Local Storage

Local Storage is a browser feature used to store data locally on the user’s machine. In this project, it stores user account details and balance without a backend. Data remains even after the browser is refreshed or closed. It’s simple to use for small-scale applications like prototypes or demos.

4. IDE: Visual Studio Code

VS Code is a lightweight and powerful code editor for web development. It supports JavaScript and React with features like IntelliSense and debugging. Extensions like ESLint, Prettier, and React snippets improve productivity. Its terminal integration and Git support make it developer-friendly.

5. Browser: Google Chrome / Mozilla Firefox

Modern browsers like Chrome and Firefox are used to test and run the web application. They support advanced JavaScript features and developer tools. Built-in consoles help debug and monitor app behavior easily. They ensure the app works smoothly across different environments.

5. System Requirements

To run this Bank Account Management System, certain software requirements must be met. The system requires **Node.js** and **npm (Node Package Manager)**, which are essential for running and managing the React.js environment. Additionally, a modern web browser like Google Chrome or Mozilla Firefox is needed to access and interact with the application smoothly.

On the hardware side, the system should run on a machine with a minimum of **4 GB RAM** to ensure smooth performance. A **processor of Intel i3** or above is recommended to handle the development and execution of the web application efficiently. These basic hardware specifications are sufficient for small-scale web projects built using React.js.

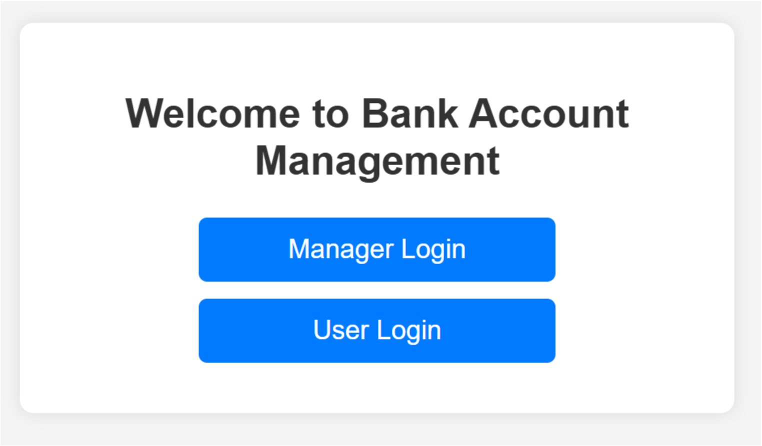
6. System Design

The architecture of this project follows a **client-side design** using React.js. The flow begins with the **user interacting with various React components** such as registration, login, and dashboard. All user information and transaction data are handled entirely on the front-end using **Local Storage** for storing account details and balance. This architecture makes the system lightweight and responsive, suitable for small-scale applications or prototypes without backend integration.

The project is organized into multiple **React components**, each handling a specific functionality. The Home.js component acts as the landing page with navigation links. Register.js allows the bank manager to register a user, and Login.js handles customer login. After login, Dashboard.js displays the user's account and transaction options. Credit.js and Debit.js manage the money addition and withdrawal processes respectively, making the system modular and easy to maintain.

7. Screenshots of Web Page

7.1 Home page of Bank Account Management System



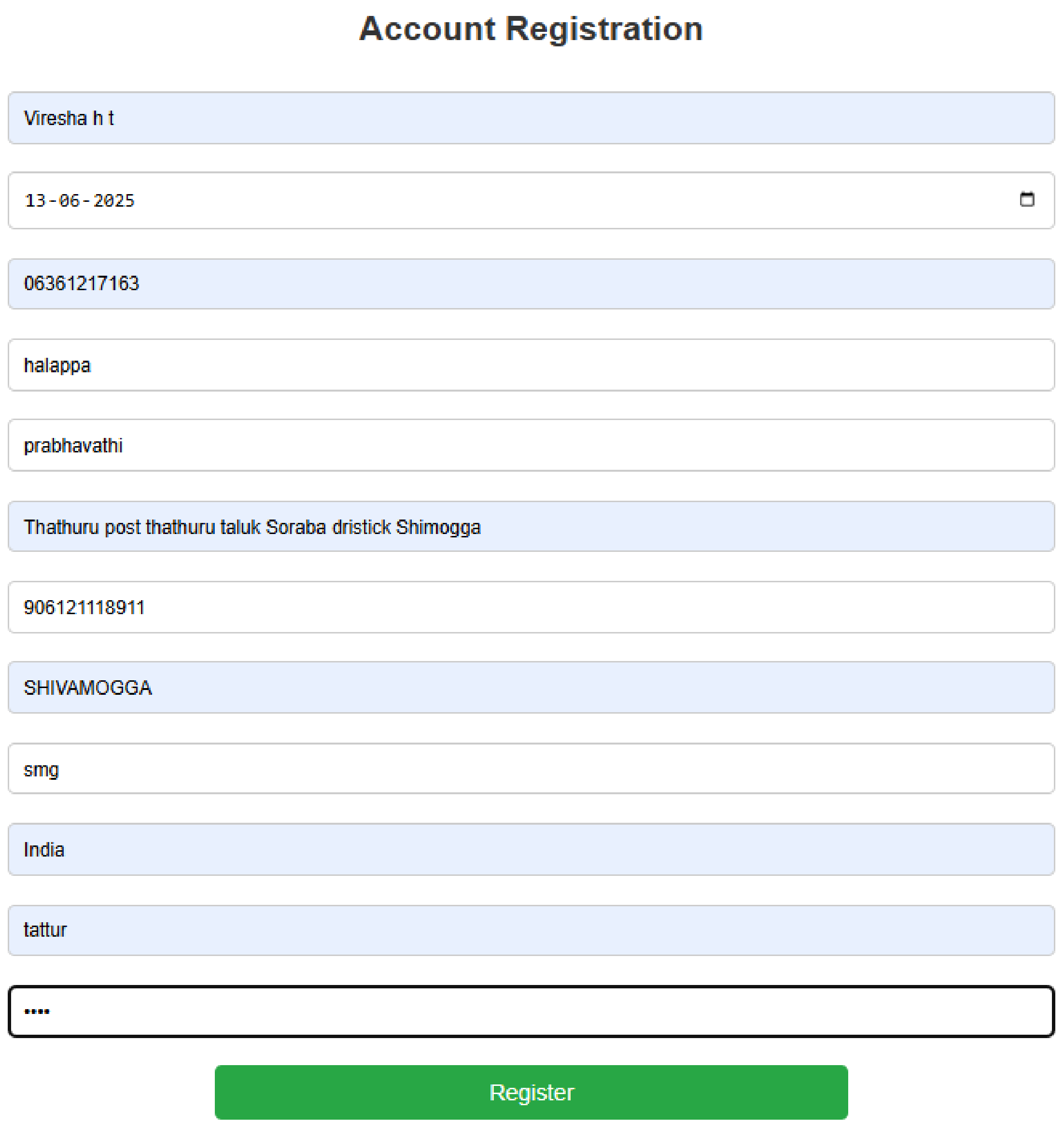
Welcome to Bank Account Management

Figure 1: Home page of Bank Account Management System

The Home page provides navigation links for account registration and user login. It serves as the entry point to the application.

7.1 Account Registration Form

This form allows the bank manager to input customer details and generate a unique account number. It collects all necessary information for account creation.



Account Registration

Viresha h t

13-06-2025

06361217163

halappa

prabhavathi

Thathuru post thathuru taluk Soraba dristick Shimogga

906121118911

SHIVAMOGGA

smg

India

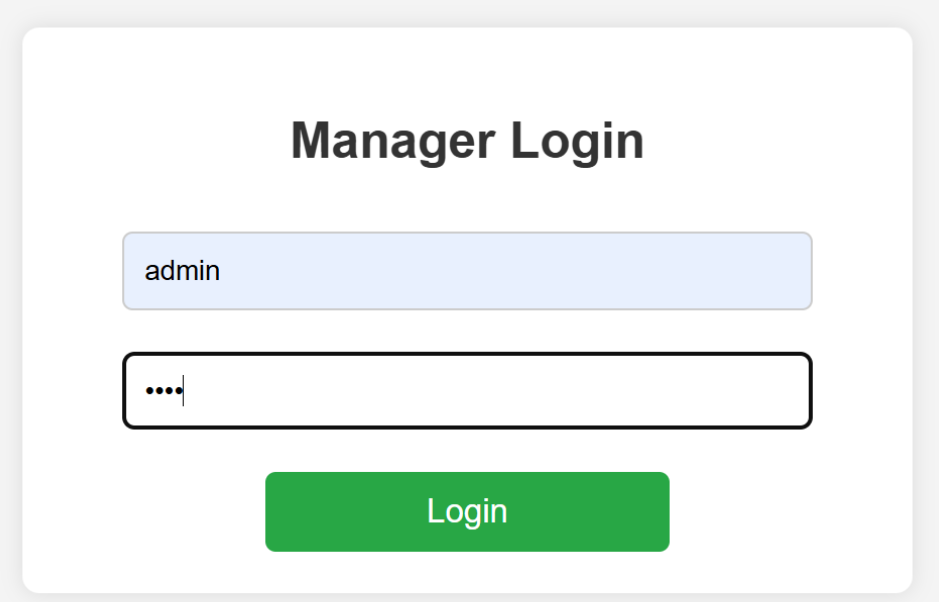
tattur

••••

Figure 2: Account Registration Form

7.3 Login Page:

Users enter their account number and password here to access their account securely. It validates credentials before granting access.

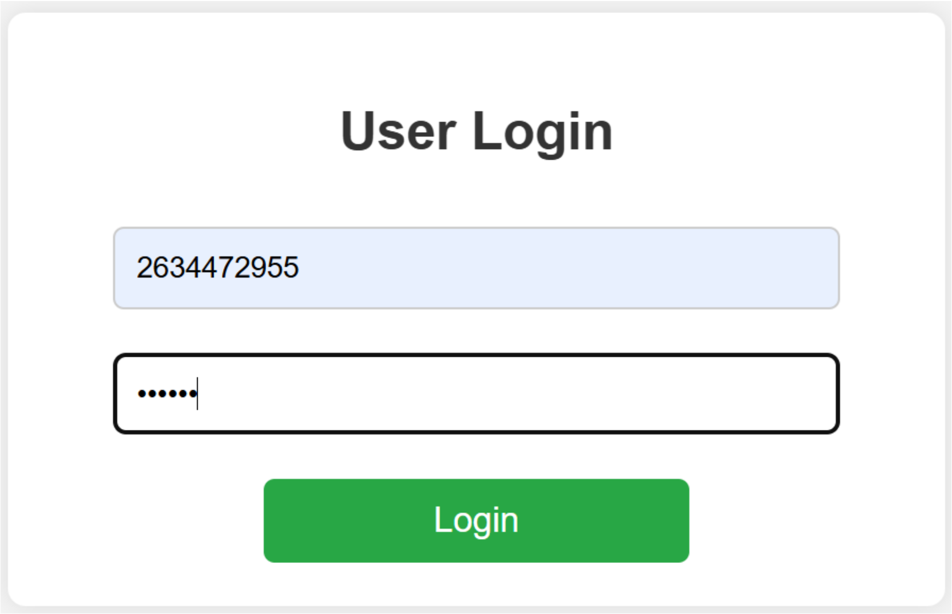


Manager Login

admin

••••

Figure 3: Manager Login Page



User Login

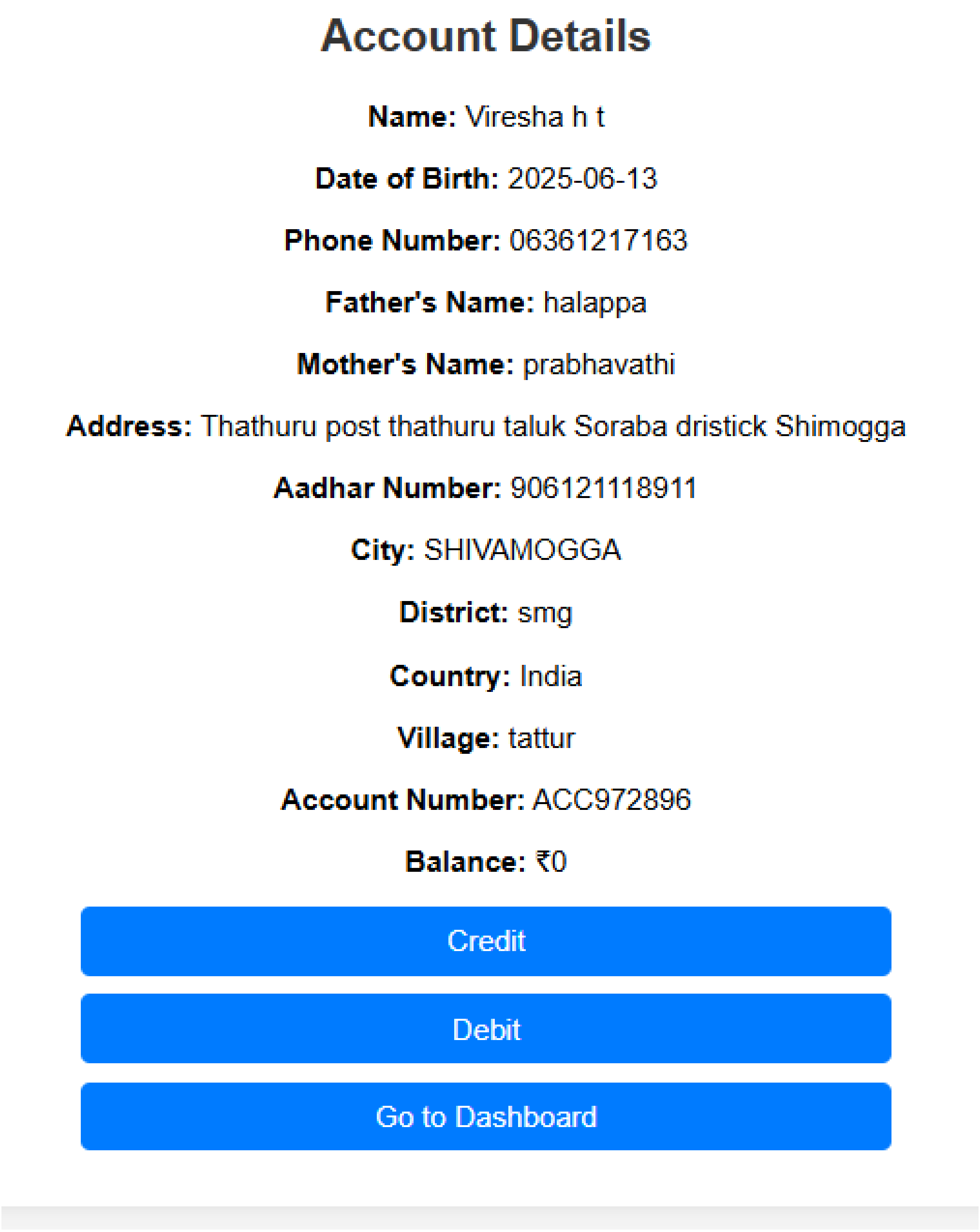
2634472955

••••••

Figure 3: User Login Page

7.4 Dashboard with Balance:

The dashboard shows the user's account information including the current balance. It also provides options for crediting and debiting the account.



Account Details

Name: Viresha h t

Date of Birth: 2025-06-13

Phone Number: 06361217163

Father's Name: halappa

Mother's Name: prabhavathi

Address: Thathuru post thathuru taluk Soraba dristick Shimogga

Aadhar Number: 906121118911

City: SHIVAMOGGA

District: smg

Country: India

Village: tattur

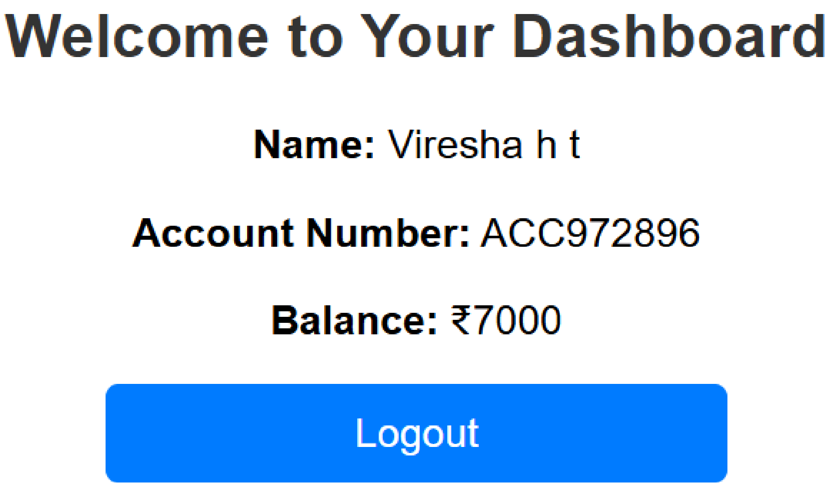
Account Number: ACC972896

Balance: ₹0

Figure 4: User Dash Board

7.5 Credit/Debit Pages:

These pages let users enter amounts to deposit or withdraw. After submission, they update and display the new account balance accordingly.



Welcome to Your Dashboard

Name: Viresha h t

Account Number: ACC972896

Balance: ₹7000

Figure 5: User Dash Board with Balance

8. Project Code:

8.1 Index.htlm

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Bank Account Management</title>

    <link rel="stylesheet" href="styles.css">

</head>
```

```
<body>
  <div class="container">
    <h2>Welcome to Bank Account Management</h2>
    <a href="manager-login.html" class="button">Manager Login</a>
    <a href="user-login.html" class="button">User Login</a>
  </div>
</body>
</html>
```

8.2 Script.js code

This is the Java script code for this project:

```
document.addEventListener("DOMContentLoaded", () => {
const $ = id => document.getElementById(id);
// Register User
if ($("registerForm")) {
$("registerForm").addEventListener("submit", e => {
e.preventDefault();
let accNo = Math.floor(1e9 + Math.random() * 9e9).toString();
let data = {
name: $("name").value,
dob: $("dob").value,
phone: $("phone").value,
father: $("fatherName").value,
mother: $("motherName").value,
addr: $("address").value,
aadhar: $("aadhar").value,
city: $("city").value,
dist: $("district").value,
country: $("country").value,
village: $("village").value,
pass: $("password").value,
accNo: accNo,
balance: 0 };
localStorage.setItem(accNo, JSON.stringify(data));
alert("Account created! No: " + accNo);
window.location.href = "details.html";
});}
// Login
if ($("loginForm")) {
$("loginForm").addEventListener("submit", e => {
e.preventDefault();
let accNo = $("accountNumber").value;
let pass = $("password").value;
let data = JSON.parse(localStorage.getItem(accNo));
if (data && data.pass === pass) {
sessionStorage.setItem("currentAcc", accNo);
window.location.href = "details.html";
} else {
```

```
alert("Invalid credentials");
}});}
// Show Account Info
if ($("accountInfo")) {
let accNo = sessionStorage.getItem("currentAcc");
let data = JSON.parse(localStorage.getItem(accNo));
$("accountInfo").innerHTML = data ? `
Name: ${data.name}<br>
Account No: ${data.accNo}<br>
Balance: ₹${data.balance}` : "No account found!";
}
});
// Navigation
function goToCreditPage() { window.location.href = "credit.html"; }
function goToDebitPage() { window.location.href = "debit.html"; }
// Credit Amount
function creditMoney() {
let accNo = sessionStorage.getItem("currentAcc");
let data = JSON.parse(localStorage.getItem(accNo));
let amt = parseFloat(document.getElementById("creditAmount").value.trim());
if (amt > 0) {
data.balance += amt;
localStorage.setItem(accNo, JSON.stringify(data));
alert(`₹${amt} credited! New Balance: ₹${data.balance}`);
```

```
window.location.href = "details.html";
} else alert("Enter valid amount");
}
// Debit Amount
function debitMoney() {
let accNo = sessionStorage.getItem("currentAcc");
let data = JSON.parse(localStorage.getItem(accNo));
let amt = parseFloat(document.getElementById("debitAmount").value.trim());
if (amt > 0 && data.balance >= amt) {
data.balance -= amt;
localStorage.setItem(accNo, JSON.stringify(data));
alert(`₹${amt} debited! New Balance: ₹${data.balance}`);
window.location.href = "details.html";
} else alert("Invalid or insufficient amount");
}
```

9.Conclusion

This project presents a basic yet effective Bank Account Management System using React.js. It allows a bank manager to register users and enables users to log in securely. Users can view their balance and perform credit or debit operations. The system uses React Router for page navigation and local storage for data persistence. It focuses on the practical application of frontend development concepts. Important features like forms, routing, and state handling are covered. The interface is simple and user-friendly for demonstration purposes. Though limited in scope, the project builds a strong foundation for further improvements. It offers students valuable hands-on experience in developing real-world web applications. Overall, it is a useful learning tool in the domain of web development.