NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

REACT PROJECT
on
ONLINE BANKING

Under the Guidance of
Sd.Rizwana(M.Tech,
Assistant.Professor)

Presented by
A.Jashwanth (22471A05L3)
SK.I.F.Basha (22471A05P1)
Y.Leela Krishna (22471A05P2)

Signature of External Examiner

ONLINE BANKING SYSTEM

- Online banking, also known as internet banking, e-banking or virtual banking, is an electronic payment system that enables customers of a bank or other financial institution to conduct a range of financial transactions through the financial institution's website.
- To access a financial institution's online banking facility, a customer with internet access would need to register with the institution for the service, and set up a password and other credentials for customer verification.
- The types of financial transactions which a customer may transact through online banking usually includes obtaining account balances, lists of the latest transactions, electronic bill payments and funds transfers between a customer's or another's accounts.
- Modern banks not only deal in money and credit but they also perform various functions, namely, agency functions, management of foreign trade, finance, etc.
- The meaning of modern banks is used in narrow sense of the term as commercial banks.

TYPES OF INTERNET BANKING

- **Informational Websites :-** Such services are known as first level of e-banking. Through such services bank provides marketing information regarding banking products and services on a standalone server. It has very low degree of risk as there is no connection between server and bank.
- **Advanced Transactional Websites:-** A bank customer can perform non-transactional tasks through online banking, including

  Viewing account balances

  - Viewing recent transactions
  - Downloading bank statements, for example in PDF format
  - Viewing images of paid cheques
  - Ordering cheque books
  - Download periodic account statements
  - Downloading applications for M-banking, E-banking etc.

SOFTWARE AND HARDWARE REQUIREMENTS

Software Requirements

Frontend Development:

- React Framework
- HTML5, CSS3, and

Backend Development:

- Node.js with Express.js
- Database: MongoDB / MySQL

Tools & Libraries:

- Visual Studio Code
- NPM (Node Package Manager)
- API Testing: Postman

Hardware Requirements

Development Machine:

- Processor: Apple silicon chip M2
- RAM: 8GB
- Storage: 256GB SSD

Testing Devices:

- Laptop/PC for Web Application Testing
- Smartphone for Responsive UI Testing

NO OF MODULES USED IN PROJECT :

- **Frontend Development:**
- Build the user interface using HTML, CSS, JavaScript, and frameworks like React.
- Create reusable components (e.g., navbar, forms, tables, modals).
- Implement state management (e.g., Redux, Context API).
- Integrate APIs for dynamic content.
- **Backend Development:**
- Set up a server using frameworks like Node.js (Express), Django, or Flask.
- Implement business logic, authentication, and authorization.
- Develop RESTful or GraphQL APIs for interaction with the frontend.
- **Database Setup and Integration:**
- Create tables and relationships in MySQL.
- Write optimized queries (CRUD operations).
- Ensure data security (encryption, validation).

Example Workflow Using MySQL:

Frontend (React):

1. A form allows users to enter data.
2. Upon submission, data is sent to the backend via an API.

Backend (Node.js + Express):

1. Receives data, validates it, and interacts with the MySQL database.
2. Example: Stores user details in a "users" table.

Database (MySQL):

1. Stores data in structured tables.

Return Data to Frontend:

Backend sends a response to the frontend to update the UI.

Overview of Frontend, Backend, and Database

- **Frontend (React, Angular, etc.):**
  - Provides the user interface (UI) for interacting with the application.
  - Sends requests to the backend and displays the data received.
- **Backend (Node.js, Django, etc.):**
  - Processes requests from the frontend and performs logic.
  - Connects to the database to perform CRUD operations (Create, Read, Update, Delete).
- **MySQL Database:**
  - Stores the application data in structured tables.
  - CRUD operations interact with the data.

4. Frontend and Backend Communication

- **HTTP Methods for CRUD:**
  - **Create:** POST - To send new data to the backend.
  - **Read:** GET - To fetch data.
  - **Update:** PUT/PATCH - To modify existing data.
  - **Delete:** DELETE - To remove data.
- **API Requests:**
  - Frontend sends HTTP requests to the backend with data in the request body (for POST/PUT).
  - Backend sends responses back with status codes and data.

5. Connecting the Backend to MySQL

Using Node.js:

- **Install dependencies:**

```
npm install express mysql2 body-parser
```

- **Set up a connection:**

```
const mysql = require('mysql2’);
const connection = mysql.createConnection({
    host: 'localhost’,
    user: 'root’,
    password: 'yourpassword’,
   database: 'yourdatabase’
 });
```

SAMPLE CODE:

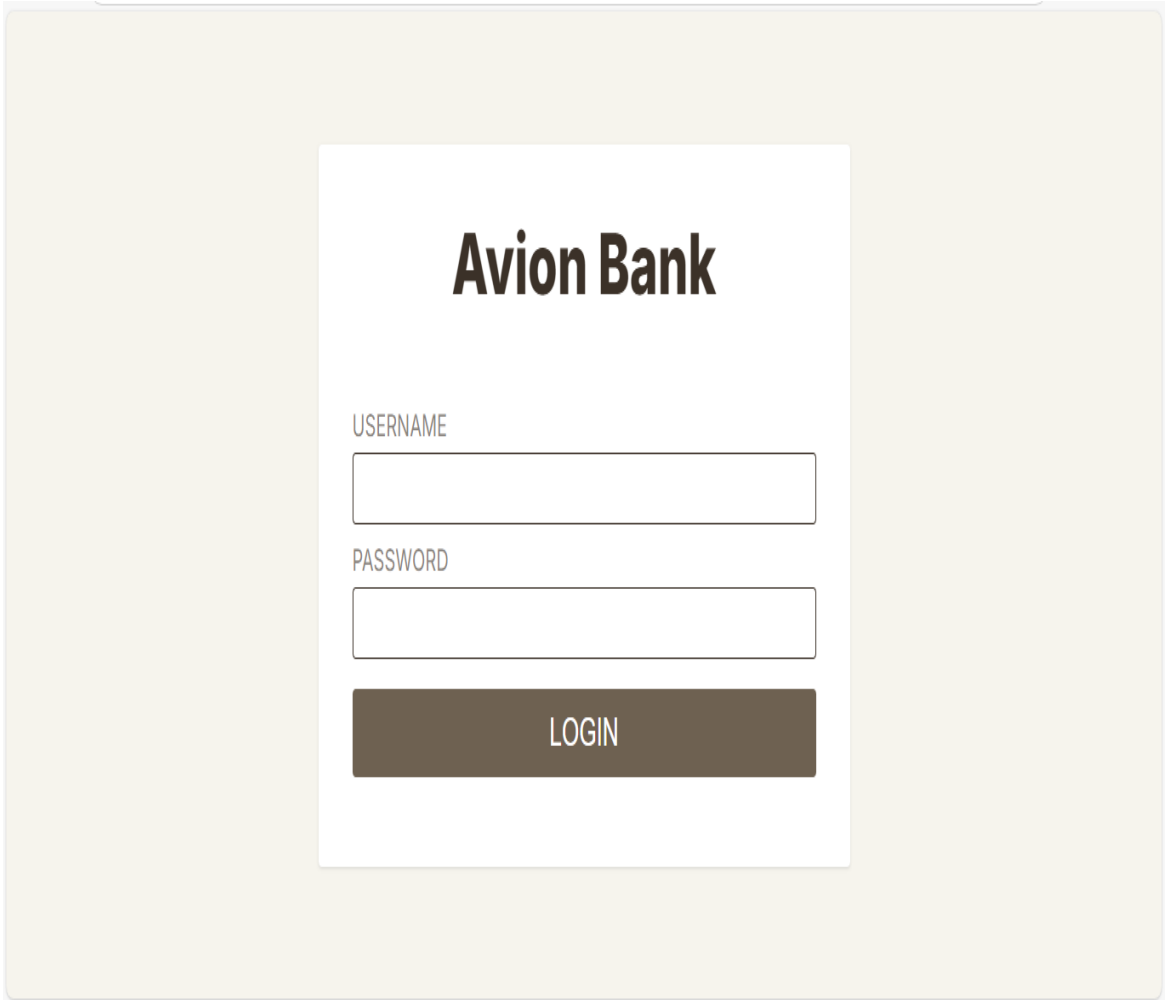
```
const express = require('express');
var cors = require('cors');
const connection = require('./connection');
const userRoute = require('./user');
const app =express();
app.use(cors());
app.use(express.urlencoded({ extended: true }));
app.use(express.json());
app.use('/user', userRoute);
module.exports = app;
```

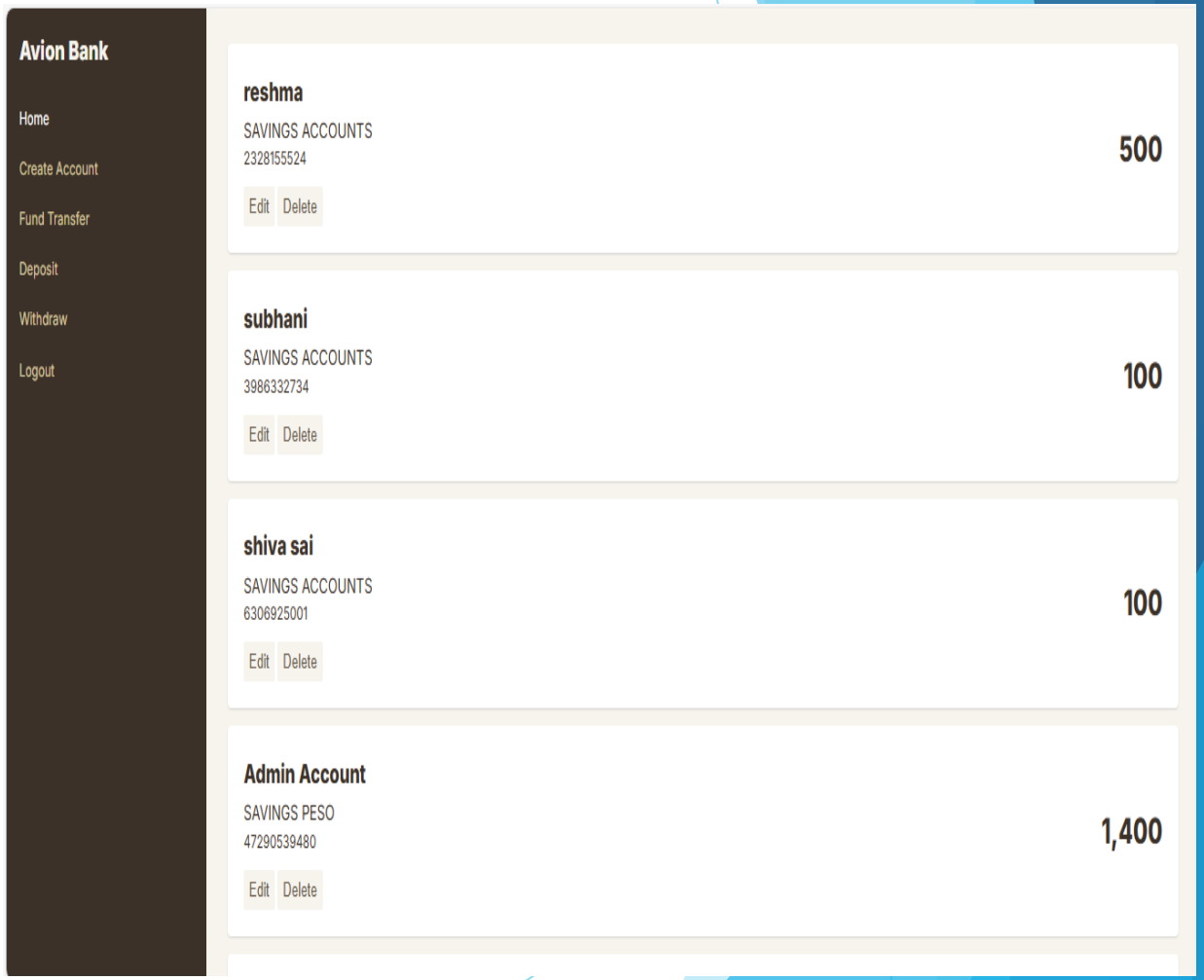
```
//server
PORT  = 8080
//connection
DB_PORT = 3306
DB_HOST = localhost
DB_USERNAME = root
DB_PASSWORD = 123456
DB_NAME = user
```

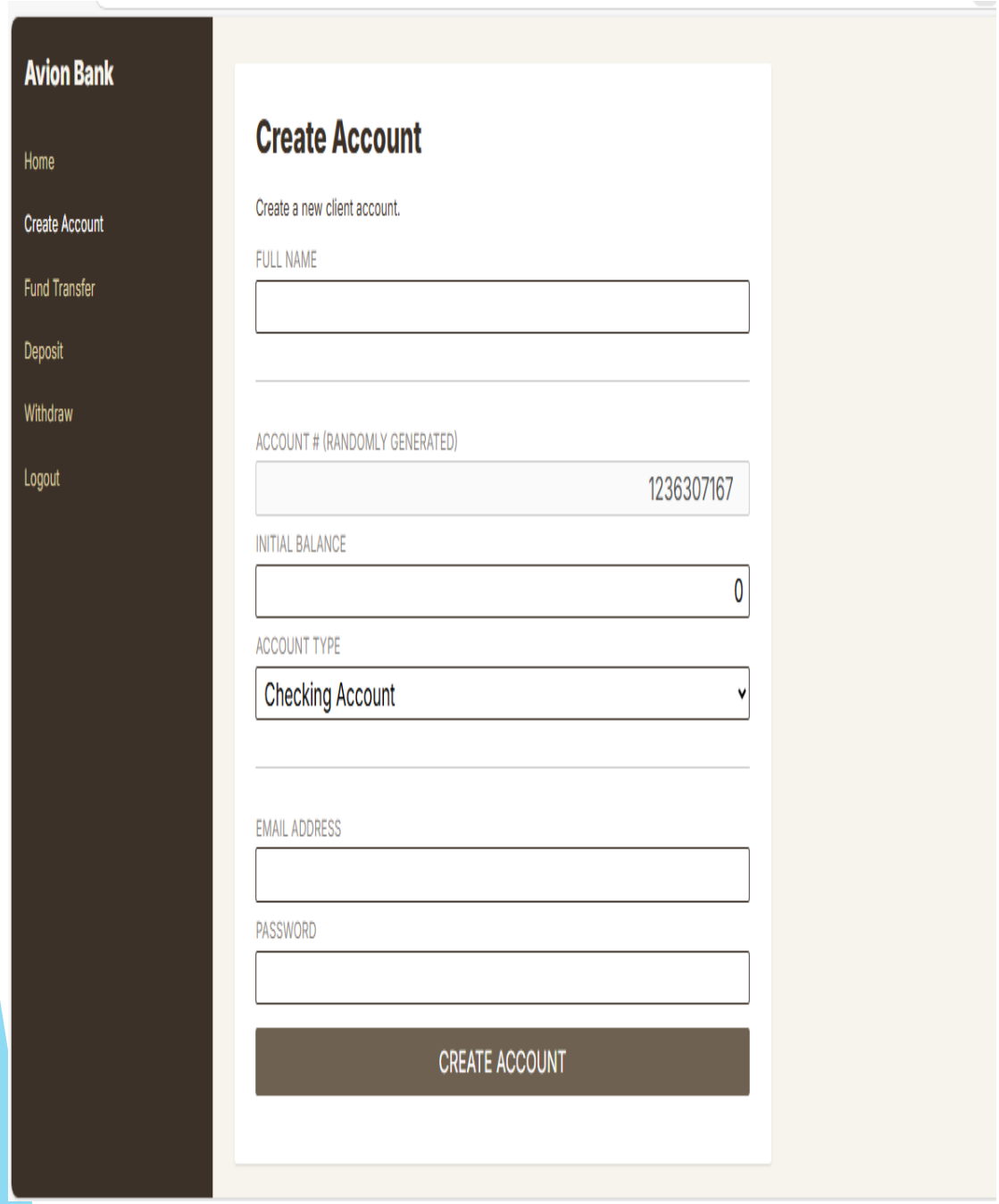
```
require('dotenv').config();
const http = require('http');
const app = require('./index');
const server = http.createServer(app);
server.listen(process.env.PORT, () => {
console.log(`Server is running on port ${process.env.PORT}`);
});
```

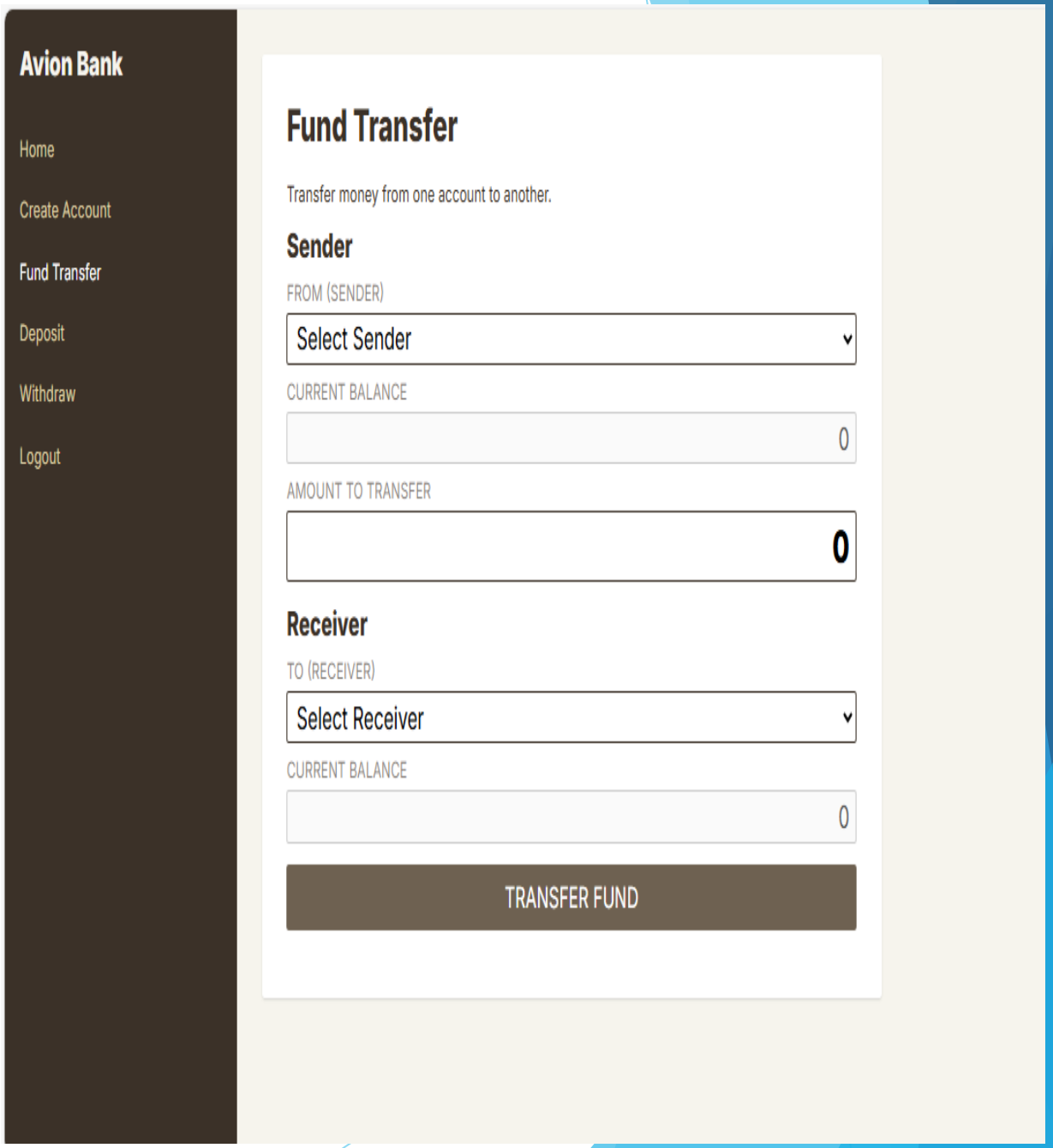
```
const mysql = require('mysql')
require('dotenv').config();

var connection = mysql.createConnection({
port:process.env.DB_PORT,
host:process.env.DB_HOST,
user:process.env.DB_USERNAME,
password:process.env.DB_PASSWORD,
database:process.env.DB_NAME,
});
```

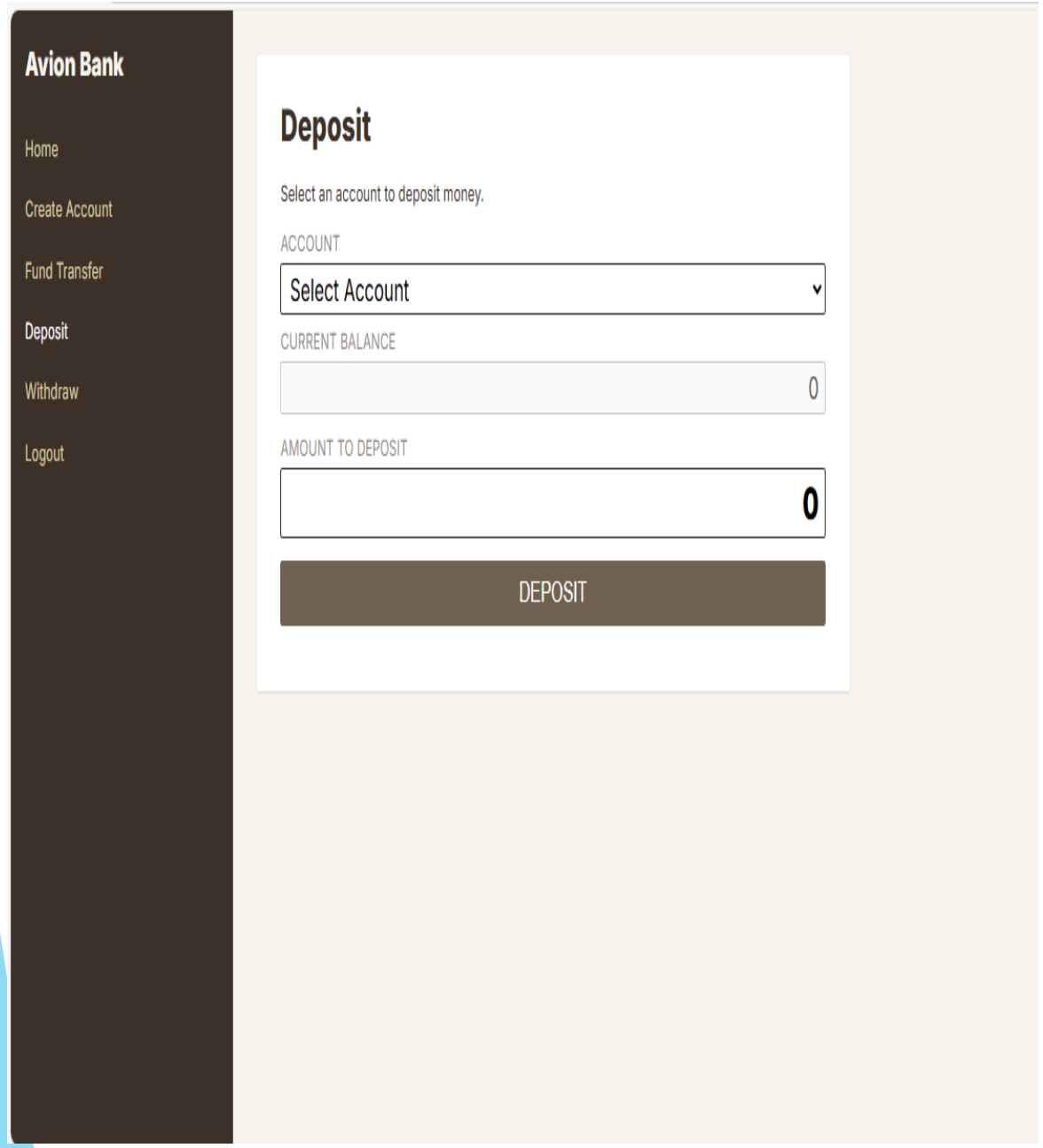
PROJECT PHOTOS:

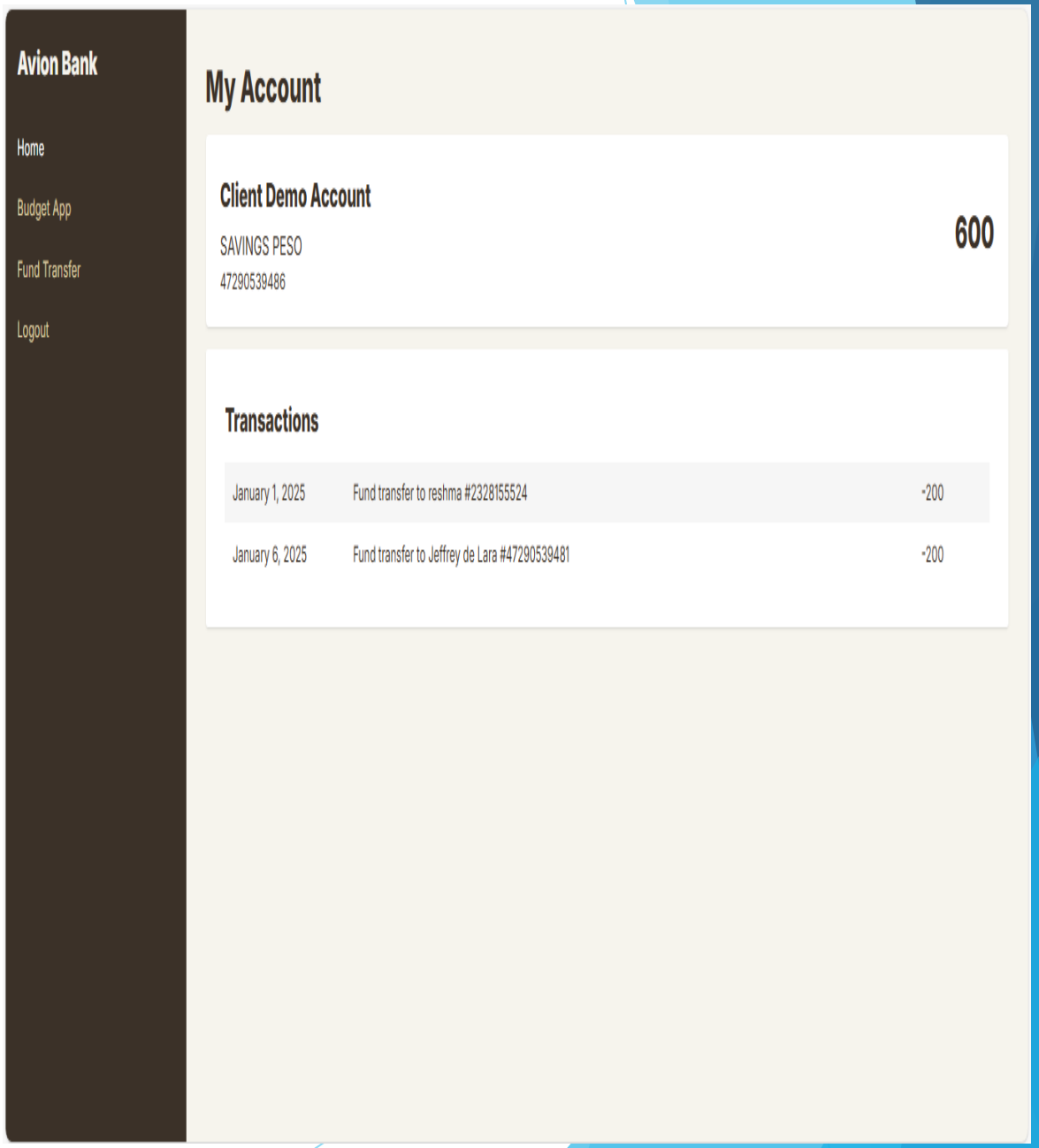





Avion Bank
Home
Create Account
Fund Transfer
Deposit
Withdraw
Logout
Create Account
Create a new client account.
FULL NAME
ACCOUNT # (RANDOMLY GENERATED)
1236307167
INITIAL BALANCE
0
ACCOUNT TYPE
Checking Account
EMAIL ADDRESS
PASSWORD
CREATE ACCOUNT


Avion Bank
Home
Create Account
Fund Transfer
Deposit
Withdraw
Logout
Fund Transfer
Transfer money from one account to another.
Sender
FROM (SENDER)
Select Sender
CURRENT BALANCE
0
AMOUNT TO TRANSFER
0
Receiver
TO (RECEIVER)
Select Receiver
CURRENT BALANCE
0
TRANSFER FUND


Avion Bank
Home
Create Account
Fund Transfer
Deposit
Withdraw
Logout
Deposit
Select an account to deposit money.
ACCOUNT
Select Account
CURRENT BALANCE
0
AMOUNT TO DEPOSIT
0
DEPOSIT


Avion Bank
Home
Budget App
Fund Transfer
Logout
My Account
Client Demo Account
SAVINGS PESO
47290539486
600
Transactions
January 1, 2025
Fund transfer to reshma #2328155524
-200
January 6, 2025
Fund transfer to Jeffrey de Lara #47290539481
-200

THANKING YOU

FROM TEAM

A.Jashwanth (22471A05L3)

SK.I.F.Basha (22471A05P1)

Y.Leela Krishna (22471A05P2)