# Mini Banking Project Documentation

## Introduction

In today’s fast-paced digital world, managing finances securely and efficiently is more important than ever. \*\*Mini Banking\*\* provides a reliable and secure banking system designed to meet modern financial needs. Whether you’re opening an account, making transactions, or managing funds, our platform ensures seamless banking operations with advanced security and scalability.

## Features

- Secure account management

- Real-time transactions

- User authentication

- Data encryption

- Compliance with security standards

## Technical Overview

### Backend Architecture

Mini Banking is built with a \*\*Node.js\*\* backend and \*\*MongoDB\*\* as the database. The backend consists of multiple APIs for handling banking operations securely.

### APIs

1. \*\*User API\*\*: Handles user registration and authentication.

2. \*\*Account API\*\*: Creates and manages user accounts.

3. \*\*Balance API\*\*: Fetches the user’s account balance.

4. \*\*Deposit API\*\*: Allows users to deposit money.

5. \*\*Withdraw API\*\*: Enables money withdrawal from accounts.

## Folder Structure

### Given Project Structure:

SERVER  
|-- controller  
| |-- bankController.js  
| |-- userController.js  
|  
|-- models  
| |-- account.js  
| |-- transaction.js  
| |-- user.js  
|  
|-- routes  
| |-- bankRoute.js  
| |-- userRouter.js  
|  
|-- node\_modules  
|-- .env  
|-- index.js  
|-- package.json  
|-- package-lock.json

### Your Modified Project Structure:

mini-banking  
|-- node\_modules  
|-- Account.js  
|-- accountRoutes.js  
|-- auth.js  
|-- mon.env  
|-- package.json  
|-- package-lock.json  
|-- server.js  
|-- user.js  
|-- userrou  
|-- userRoutes.js

## Implementation Code

### server.js

const express = require('express');  
const mongoose = require('mongoose');  
require('dotenv').config();  
  
const app = express();  
const PORT = 3000;  
  
mongoose.connect(process.env.MONGO\_URI, {  
 useNewUrlParser: true,  
 useUnifiedTopology: true  
}).then(() => console.log('✅ MongoDB connected successfully'))  
 .catch(err => console.log(err));  
  
app.listen(PORT, () => {  
 console.log(`✅ Server running on port ${PORT}`);  
});

## Execution Screenshot

Below is the screenshot showing successful execution of the server:

## Usage Flow

1. \*\*User Signup\*\*: Users register and create an account.

2. \*\*Account Management\*\*: Users can deposit, withdraw, and check balance.

3. \*\*Security\*\*: Secure authentication and encrypted transactions.

## Conclusion

Mini Banking provides a scalable and secure banking system for individual and business financial operations. With its API-driven backend, it ensures seamless and secure transactions.

\*\*Author\*\*: RK