A PRESRNTATION ON

Banking with AVS

Ayush Verma Soumya Singh Vikas Kumar Vashisht Tiwari Vikash Rajput

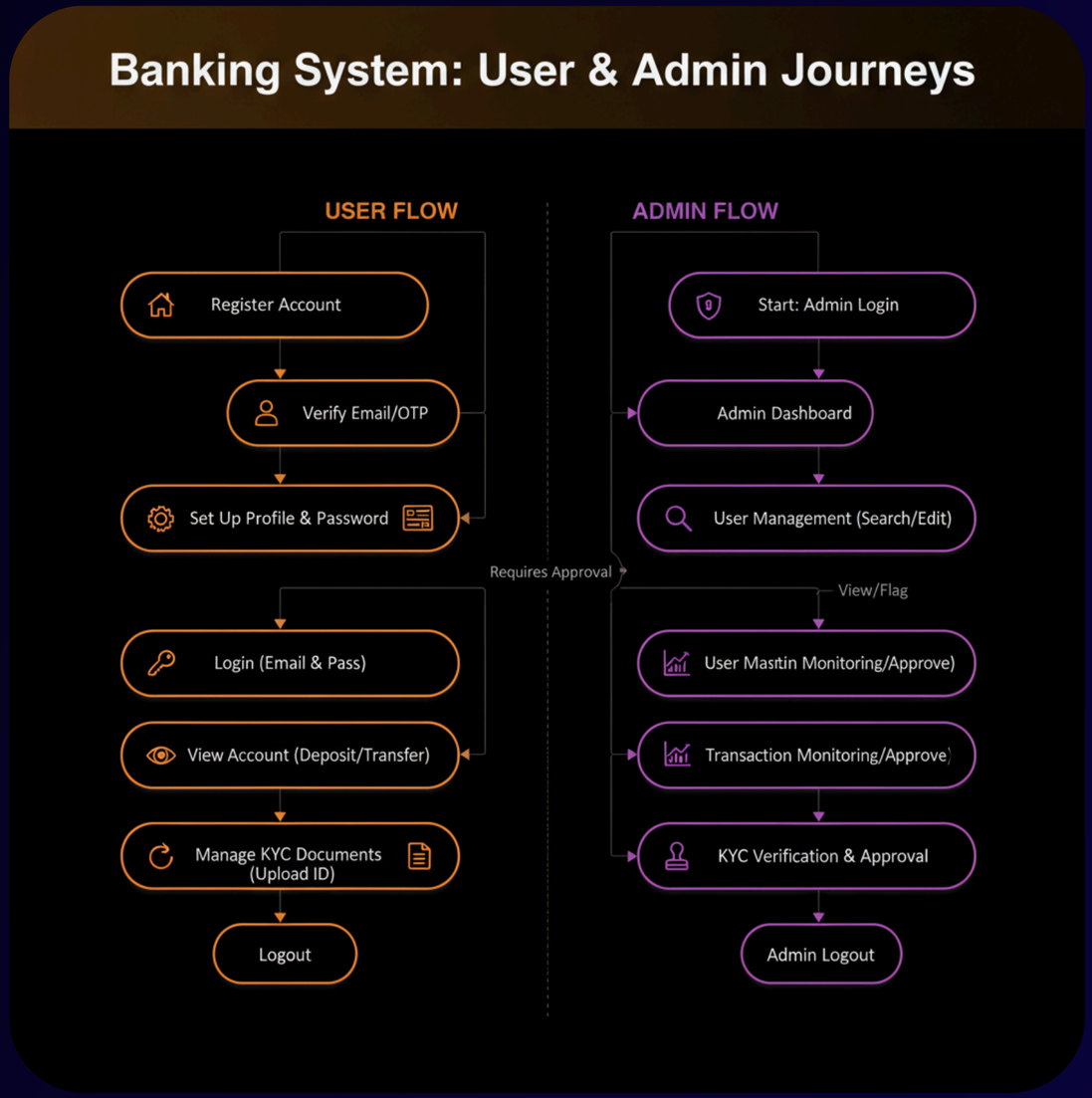
-

Table of Contents

Topics Covered

Project Overview

- AVS Bank is a full-stack banking application.
- Features user and admin roles with separate functionalities.
- Built with React (frontend) and Python Flask (backend).



Technology Stack

FRONTEND

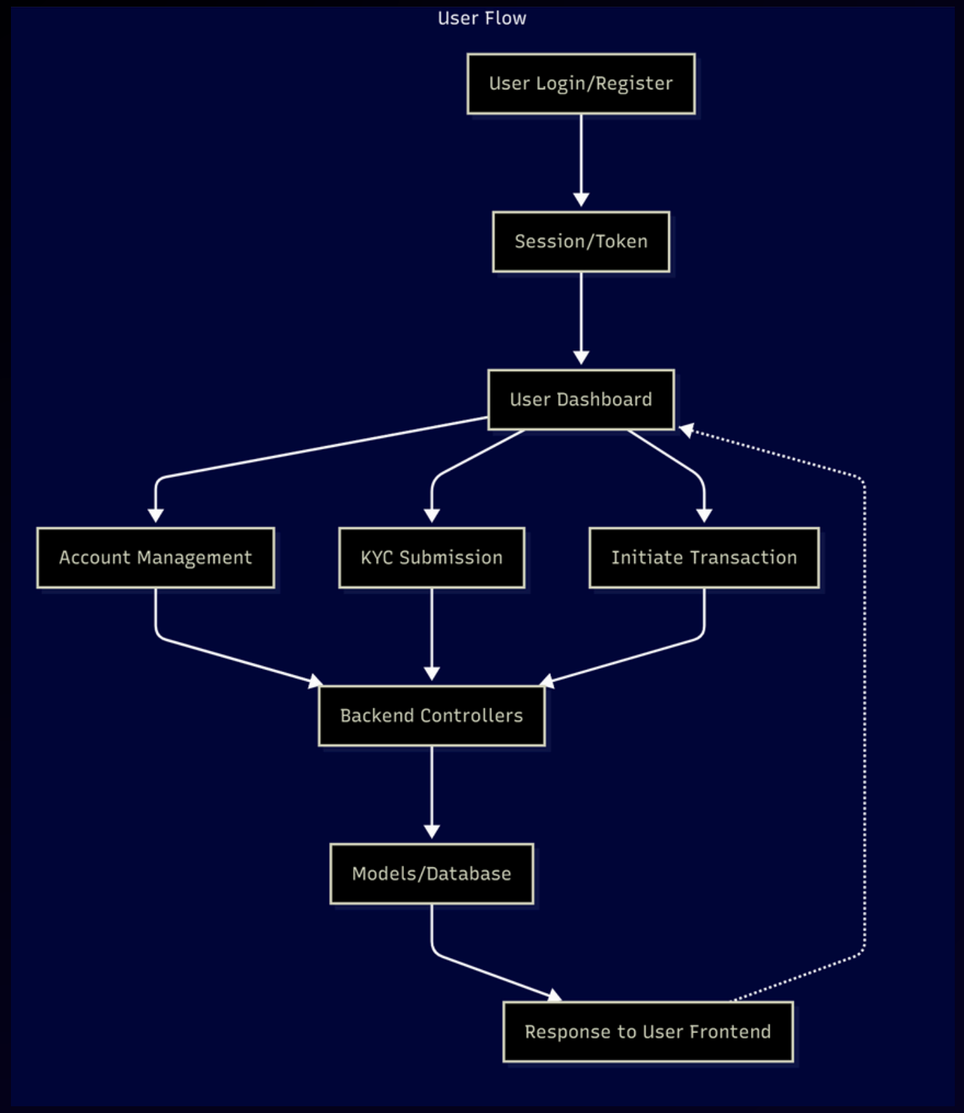
- Technology: React.js
- Routing: React Router
- API Client: Axios (for HTTP requests)

BACKEND

- **Framework: Flask** (Python)
- **API Style:** RESTful APIs
- **Authentication:** PyJWT (for JWT generation and validation)
- **ORM/Database:** SQLAlchemy (Object-Relational Mapping)
- **Serialization/Validation:** Marshmallow (Schema validation and object serialization/deserialization)

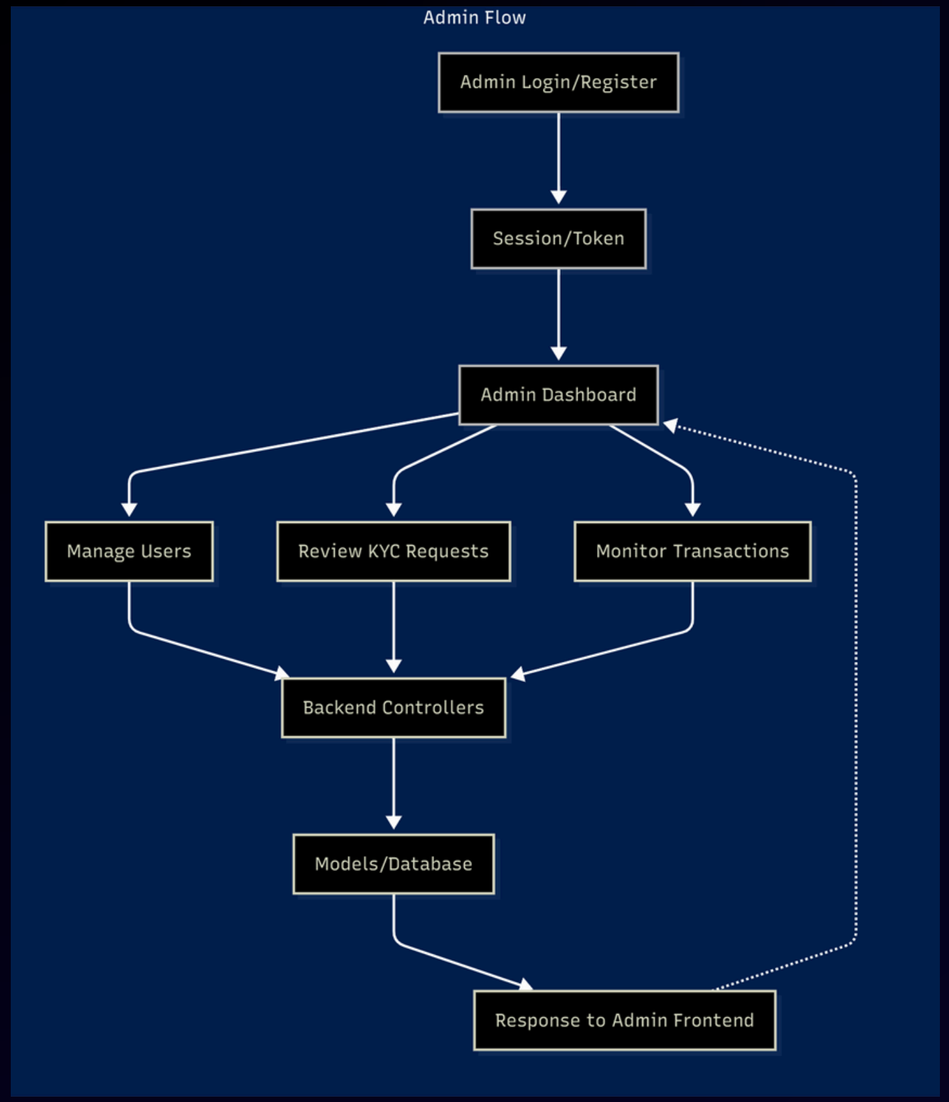
TOOLS

- **Containerization: Docker** (for application and database environment isolation)
- **CI/CD: GitHub Actions** (or GitLab CI)
- **Postman** : for checking endpoints



User Workflow

- Registration: User registers on the system, undergoing client-side and server-side validation.
- Account Creation: Successful registration creates the customer record and primary account.
- Login & Authentication: User logs in, is authenticated, and is issued a JWT (Access Key).
- Dashboard Access: The JWT grants access to the User Dashboard, the main operational hub.
- Profile Management: User can view and update their profile and account details.
- KYC Compliance: User securely uploads KYC documents, which are validated, and the system updates the KYC status.
- Core Banking: User performs deposits, withdrawals, and transfers, resulting in an updated account balance.
- Logout: User securely logs out.



Admin Workflow

- Secure Login: Admin logs into a restricted, secure portal.
- User Management: View, search, and manage customer profiles.
- KYC Verification: Review, approve, or reject uploaded KYC documents.
- Access Control (RBAC): Permissions are strictly controlled.
- Monitoring/Logging: View system logs for stability (sensitive data is masked).
- Logout: Securely exit the system.

MVC Architecture Overview

MODEL

- Handles data and business logic.
- Example: models.py, adminmodel.py, transactionmodel.py.

VIEW

- User interface components.
- Example: React pages like UserDashboard.js, AdminDashboard.js, KYC.js.

CONTROLLER

- Manages communication between View and Model.
- Example: Flask controllers like user_controller.py, admin_controller.py.

Key Implementations

1. Security Implementation
2. Validation Strategy
3. CI/CD Pipeline

Where JWT is Used in AVS Bank ?

Benefits:

- Stateless authentication (no need to store session data on the server).
- Secure and scalable.
- Enables role-based access control.

- **User Login**: After successful login, a JWT is issued to the user.
- **Admin Login**: Admins receive a JWT upon authentication.
- **Protected Routes**: Any API endpoint that requires authentication checks for a valid JWT (e.g., accessing dashboards, submitting KYC, initiating transactions).
- **Authorization**: The backend uses JWT to determine user roles (user/admin) and permissions for specific actions.

Data Protection

- **Transport Security : HTTPS is enforced** across the board to encrypt all data in transit.
- **Logging :** Sensitive data (passwords, PII) is **masked or excluded** from all application logs.

Where Password Hashing is Used in AVS Bank ?

- **User Registration:** Passwords are hashed before being saved to the database.
- **Admin Registration:** Admin passwords are also hashed for security.
- **Login Verification:** The backend hashes the entered password and compares it to the stored hash.

bcrypt to hash passwords

Validation Strategy

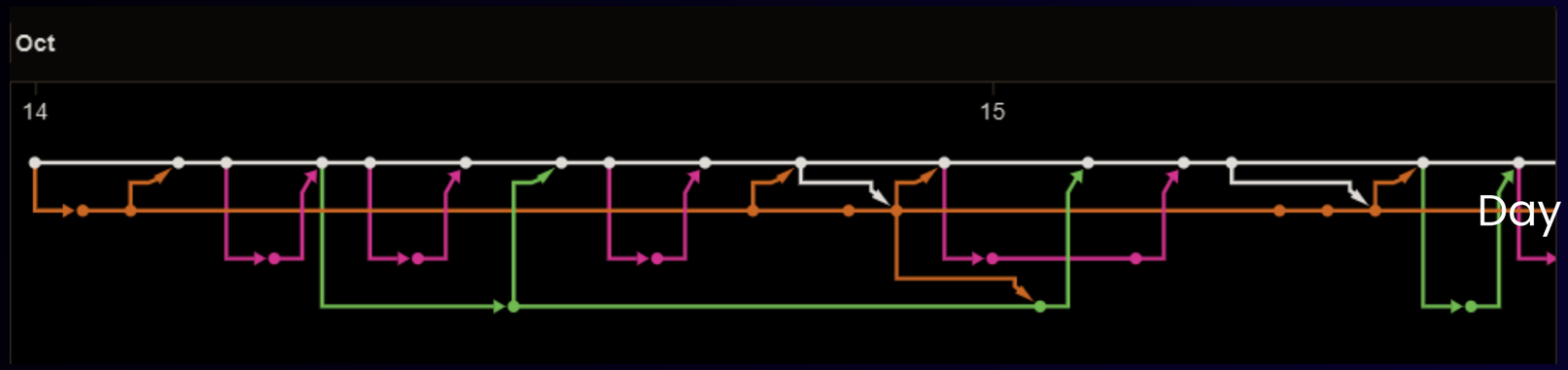
Client-Side Validations

- Required Fields: Ensures all mandatory inputs are provided.
- Length/Format: Min/max length, regex for email, phone, PAN, and Aadhaar formats.
- File Constraints: File type (PDF/JPG/JPEG) and max size (5MB) checks for uploads.
- Password Match: Confirms password and confirmation fields are identical.
- Numeric Ranges: Ensures amounts for transactions are positive and within reasonable limits.

Server-Side Validations

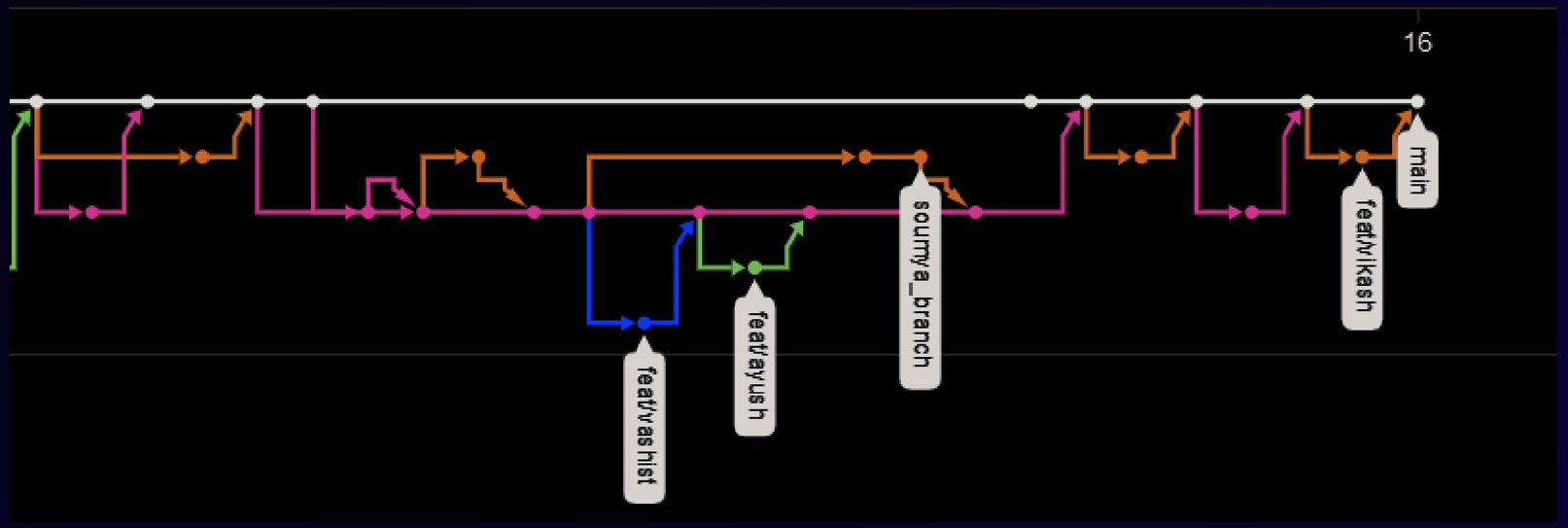
- **Re-validation: All client-side checks are repeated server-side (Never trust the client).**
- **Unique Constraints: Checks for existing email, phone, PAN, and Aadhaar values (e.g., during registration).**
- **Business Rules: Sufficient balance for withdrawals/transfers, daily transfer limits check.**
- **Security: Virus scan on all uploaded files (via an external service or library).**

CI/CD Pipeline



Day 14

Day 15



Day 16

Lighthouse and Dockerization

The application leverages **Docker** for standardized, portable, and repeatable deployment.

SCREENSHOTS

home page

User Login

User Signin

User dashboard

Admin sign in

Admin dashboard

Future Scope

Customer Support Automation:

- Deploy Conversational Chatbots: Integrate AI-powered chatbots to handle common user queries (e.g., "How do I update my address?") and guide customers through simple tasks, improving instant support.

UI/UX Refinement and Experience:

- Improving UI/UX: Conduct user testing and feedback rounds to continuously refine the visual design, interaction flows, and overall user experience.

Accessibility Focus:

- Target higher accessibility standards (WCAG 2.1 AA/AAA) to ensure the application is usable by the broadest possible audience.

Platform Expansion:

- Mobile App Expansion: Develop native or cross-platform mobile applications (iOS and Android) to provide a dedicated, fast, and feature-rich experience for users on the go.

Conclusion

This Customer Onboarding & Account Management project successfully establishes a modern, secure, and scalable digital banking module. Built on a decoupled MERN-like architecture (React/Flask) and secured with JWT and bcrypt hashing , the system ensures high data integrity through a comprehensive, two-tiered validation strategy. With a robust CI/CD pipeline and Dockerized deployment, the foundation is set for continuous improvement, ready to expand into future scopes like chatbots, mobile application development, and advanced security features to deliver an exceptional, compliant user experience.

Thank You