**Design Document for IBAN Validation System**

**Introduction**

* **Purpose**: This document provides a detailed architectural overview of the IBAN Validation System. It covers both the frontend and backend components, highlighting design choices, system architecture, and technologies used.
* **Scope**: The system allows users to register, login, and validate IBAN numbers, with valid IBANs being stored in a database. It includes admin functionalities to manage IBAN records.

**System Overview**

* **Components**:
  + **Frontend**: Developed using Vue.js, handling the user interface and interactions.
  + **Backend**: Developed using Laravel, handling API requests, data processing, and database management.
  + **Database**: MySQL, used for storing user data and IBAN validation records.

**Architectural Strategy**

* **Design Principles**:
  + **Modularity**: The frontend and backend are developed as separate modules to ensure loose coupling and high cohesion.
  + **Scalability**: System architecture supports scaling both horizontally and vertically.
  + **Security**: Emphasis on secure data handling, validated input, and secured endpoints.

**Technology Stack**

* **Frontend**:
  + **Framework**: Vue.js
  + **Routing**: Vue Router
  + **HTTP Client**: Axios for API communication
  + **Styling**: Bootstrap Vue for responsive design
* **Backend**:
  + **Framework**: Laravel
  + **Environment Management**: .env files
  + **Authentication**: Laravel Sanctum for SPA authentication
  + **Database**: MySQL
  + **ORM**: Eloquent for database interactions
* **Tools**:
  + **Version Control**: Git
  + **Package Managers**: Composer (PHP), npm /node (JavaScript)

**Database Design**

* **Schema**:
  + **Users Table**: Stores user credentials and information.
  + **IBAN Table**: Stores IBAN numbers.

**API Design**

* **Endpoints**:
  + POST /api/v1/register: Registers a new user.
  + POST /api/v1/login: Authenticates a user.
  + POST /api/v1/ibans/check: Receives IBAN, validates it, and stores if valid.
  + GET /api/v1/ users/ibans/list: Receives IBAN list with user (for admin).
  + POST /api/v1/logout: Logout user.

**Frontend Design**

* **Components**:
  + **Login Component**: Handles user login.
  + **Registration Component**: Manages user registration.
  + **IBAN Validator Component**: Allows users to input and validate IBANs.
  + **Admin Component** : Allows admin to monitor the register user and list of IBANs.

**Security Considerations**

* **Data Validation**: Server-side validation to prevent SQL injection and other malicious attacks.
* **Authentication**: Secure handling of user authentication, using tokens, csrf-cookie.

**Challenges and Solutions**

* **Cross-Origin Resource Sharing (CORS)**: Configured CORS in Laravel to accept requests from the specific Vue.js frontend.
* **IBAN Validation Logic**: Developed custom validation logic instead of relying on third-party libraries some research
* **Laravel version 11x (some customization issues)**

**Conclusion**

* Summary of the system capabilities and the design choices that enable these capabilities.

**Appendices**

**C. References**:

https://laravel.com/docs/11.x/installation

https://vuejs.org/guide/introduction.html

<https://www.iban.com>  
<https://chatgpt.com>

http://youtube.com

System admin can create from MYSQL

Sample Admin :

Security credentials

Email ->[admin@gmail.com](mailto:admin@gmail.com)

Password -> Admin@1234

**Note : backend run in this port**

[**http://127.0.0.1:8000**](http://127.0.0.1:8000)

**frontend .env config is not working**

**Run the project using (local)**

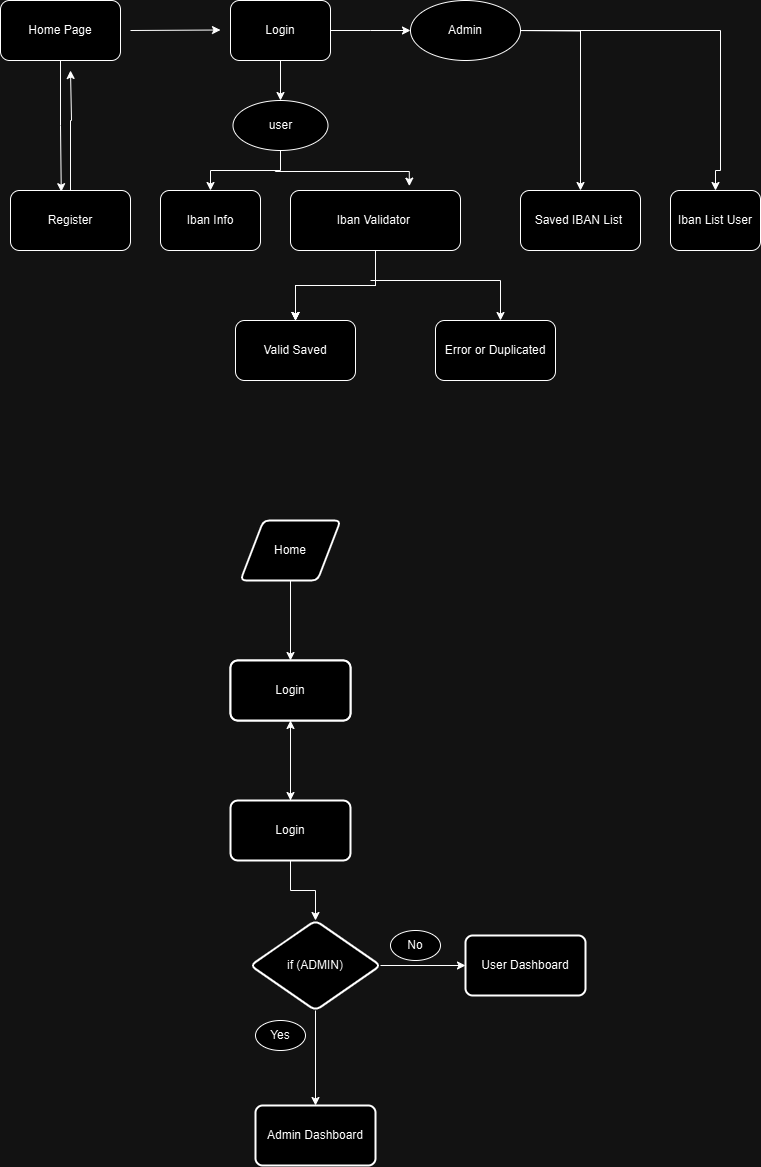
**npm run dev**

**or**

**npm run serve**

**Project System Design**

**code design**

****