**Software Architecture – High-Level Design**

**1. Overview**

This document outlines the software architecture for the system, detailing its components, interactions, and architecture style to ensure scalability, maintainability, and efficiency.

**2. Architecture Style**

The system follows the **Model-View-Controller (MVC)** architecture to separate concerns and enhance modularity.

* **Model (Data Layer)**:
  + Represents the core data logic and database interactions.
  + Manages entities such as **User, Donation, Payment, and Receipt**.
  + Uses SQL Server for efficient data storage and retrieval.
* **View (Presentation Layer)**:
  + Responsible for displaying data and handling user interactions.
  + Developed using HTML, CSS, JavaScript, and Bootstrap for a responsive design.
* **Controller (Business Logic Layer)**:
  + Handles requests from users, processes business logic, and communicates with the database.
  + Manages authentication, validation, and API interactions.

**3. System Components**

**3.1. User Interface (Frontend)**

* Handles user interactions and data input.
* Sends requests to the backend for processing.

**3.2. Backend (Business Logic Layer)**

* Implements the system’s core logic and processes user requests.
* Communicates with the database and external services.

**3.3. Database Layer**

* Stores user information, donation records, and authentication data.
* Ensures data integrity and efficient querying.

**3.4. Authentication Module**

* Implements **JWT (JSON Web Token)** for secure user authentication and access control.
* Validates user credentials before allowing access to protected resources.

**3.5. Payment Integration Module**

* Manages secure transactions using third-party payment gateways (e.g., PayPal, Stripe).
* Ensures seamless processing of online donations.

**3.6. Notification System**

* Sends email confirmations and alerts for donation status and account activity.
* Ensures timely communication between users and the system.

**4. System Interactions**

1. **User Authentication**: The donor or admin logs in, and the system verifies credentials using JWT.
2. **Donation Processing**: The donor selects a donation category, enters payment details, and submits the donation.
3. **Payment Gateway Integration**: The system interacts with third-party payment APIs to process transactions securely.
4. **Receipt Generation & Notifications**: After a successful donation, a receipt is generated and sent via email.
5. **Admin Management**: The administrator monitors donations, manages users, and generates reports through the admin dashboard.

This structured approach ensures the system is scalable, maintainable, and secure, providing a seamless experience for both donors and administrators.