create database charity;

USE charity;

CREATE TABLE Volunteer (

    volunteer\_id INT PRIMARY KEY AUTO\_INCREMENT NOT NULL,

    first\_name VARCHAR (100) NOT NULL,

    last\_name VARCHAR (100) NOT NULL,

    age INT NOT NULL,

    email VARCHAR (100) NOT NULL,

    mobile INT NOT NULL,

    city VARCHAR (100) NOT NULL,

    country VARCHAR (100) NOT NULL

);

CREATE TABLE Donation (

    donation\_id INT PRIMARY KEY AUTO\_INCREMENT NOT NULL,

    first\_name VARCHAR (100) NOT NULL,

    last\_name VARCHAR (100) NOT NULL,

    email VARCHAR (100) NOT NULL,

    mobile INT NOT NULL,

    amount INT NOT NULL,

    card\_number LONG NOT NULL,

    donation\_date DATE NOT NULL DEFAULT (current\_date())

   );

 CREATE TABLE Contact (

    contact\_id INT PRIMARY KEY AUTO\_INCREMENT NOT NULL,

    first\_name VARCHAR (100) NOT NULL,

    last\_name VARCHAR (100) NOT NULL,

    email VARCHAR (100) NOT NULL,

    messages VARCHAR(100) NOT NULL

);

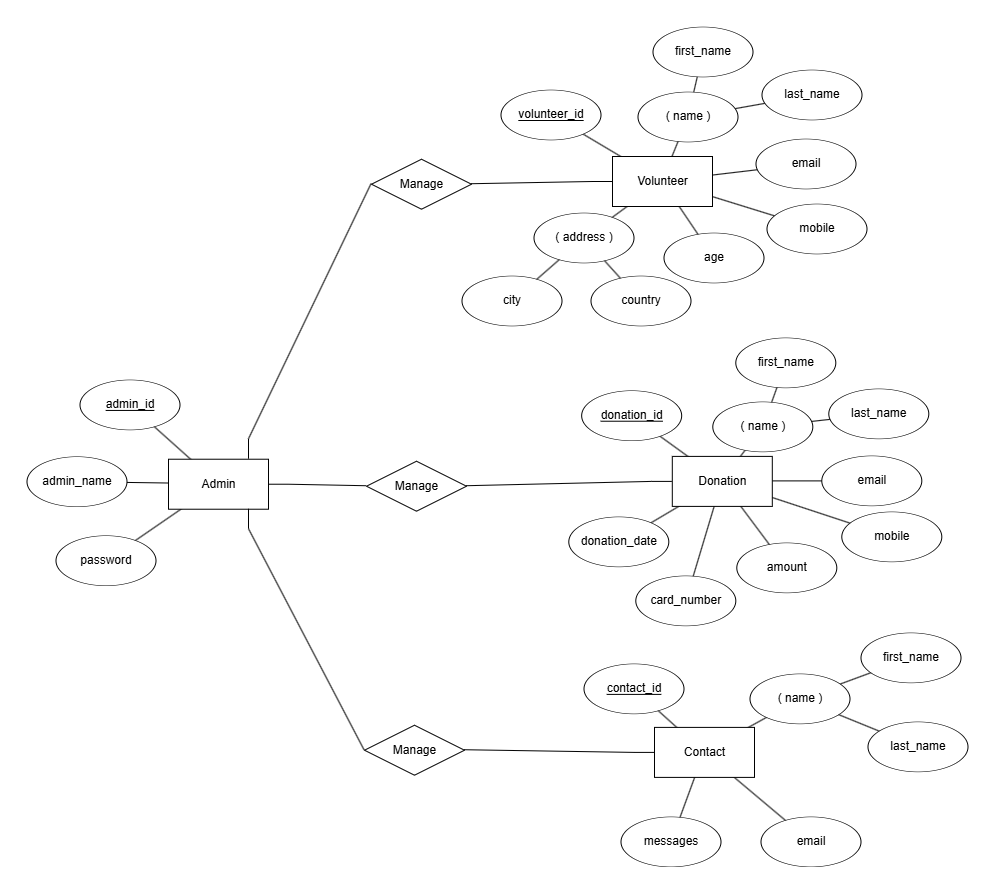
CREATE TABLE Admin (

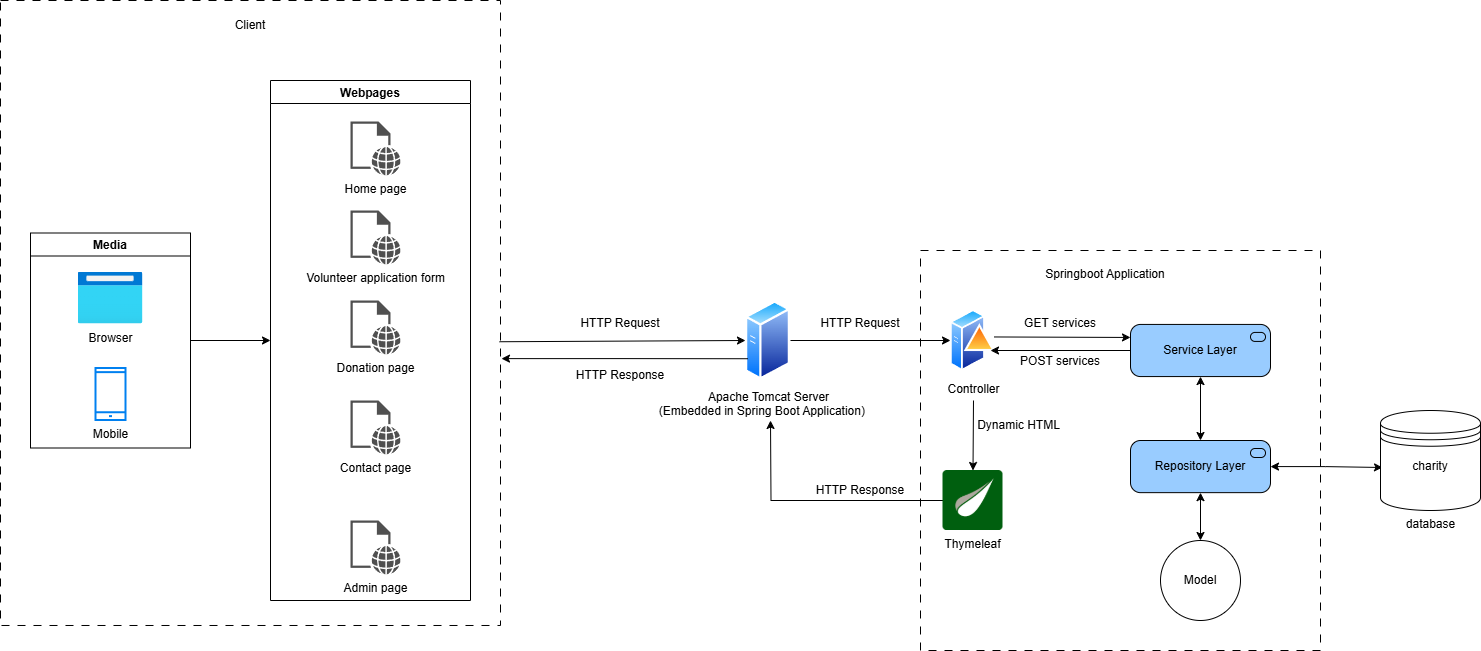
    admin\_id INT PRIMARY KEY AUTO\_INCREMENT NOT NULL,

    admin\_name VARCHAR (100) NOT NULL,

    password VARCHAR(100) NOT NULL

);

INSERT INTO `charity`.`admin` (`admin\_id`, `admin\_name`, `password`) VALUES ('1', 'admin', 'password');



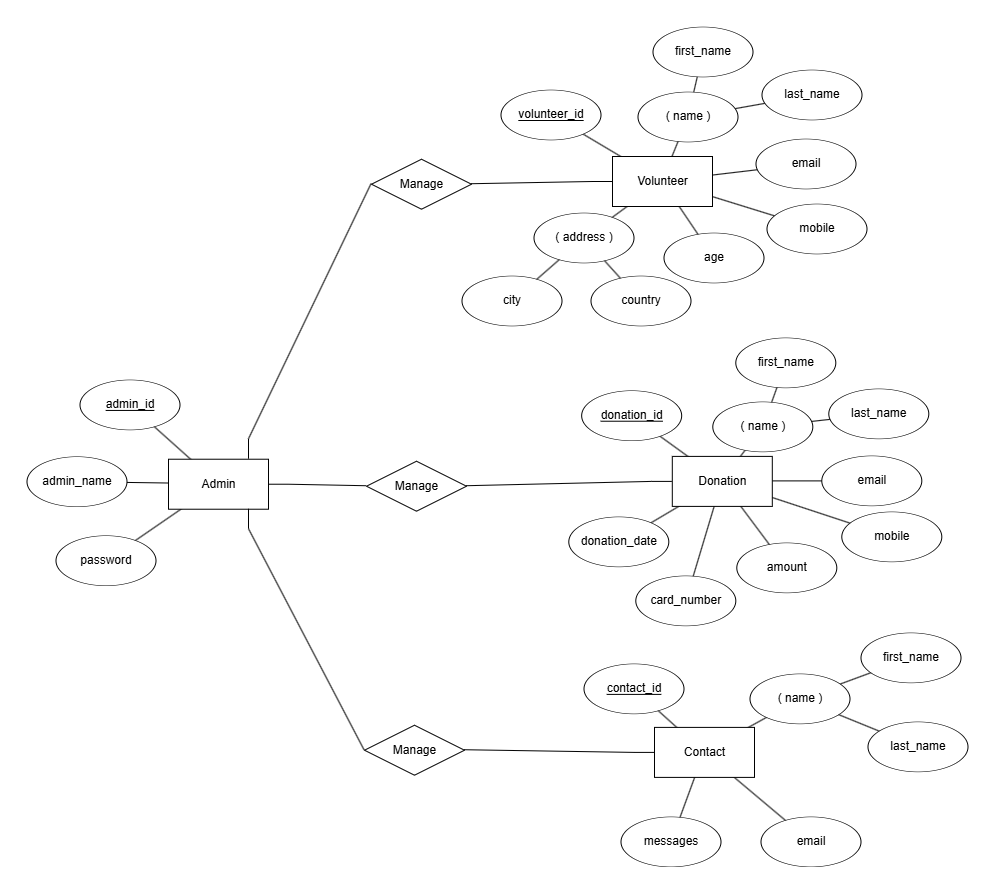
Spring Boot is a framework that simplifies the creation of Spring-based Java applications by providing an opinionated setup with minimal configuration, using an embedded Apache Tomcat server. Thymeleaf is a Java-based template engine designed for generating views and is used in the presentation layer to dynamically render HTML and other text-based formats, with HTML-friendly syntax and tight integration with the Spring Framework. While Spring Boot handles the backend logic, routing, and services, Thymeleaf is used to render dynamic HTML pages on the frontend. They complement each other but are not interchangeable.

The system architecture design diagram above outlines how the front-end, back-end, and database components work together.  First, the client is end-user interacts with the webpages, including home page, volunteer page, contact page, donation page and admin page via a web browser or mobile interface.The Apache Tomcaet Server is embended in the Spring Boot application to process incoming HTTP requests. When a client sends an HTTP request, it is routed to the appropriate controllers within the Spring Boot application.

In the Spring Boot application,the architecture's backbone includes the controller that handles HTTP requests, the service layer containing business logic and rules, the repository layer communicating with the database to fetch or store data, and the model representing the application's data structure and business objects.

The workflow in the Spring Boot application begins when the controller receives an HTTP request, delegating business logic to the service layer. The service layer interacts with the database via the repository layer to fetch or store data. The retrieved data is then sent to Thymeleaf templates for rendering, dynamically generating an HTML response based on the template and data. The generated HTML, also an HTTP response, is sent back to the client via the Apache Tomcat Server, and the client displays the rendered HTML page.

In terms of database interaction, when a controller requests service from the service layer, the service layer calls the repository layer to interact directly with the database and retrieve data. The repository layer translates repository methods into SQL queries and executes them against the database. A model defines how the data should be stored, retrieved from the database, and represented within the application. After retrieving the database rows, the repository layer maps them to the corresponding model using Hibernate to represent the data in the application. The represented data is then returned to the service layer and finally passed to the controller.

The database structure is shown in the ERD diagram above which is designed to manage a system involving administrators, volunteers, donation records, and contact messages. The core entity, Admin, is responsible for managing all other entities, including Volunteer, Donation, and Contact. Each admin is uniquely identified by an admin\_id and has credentials such as a password and admin\_name for authentication purposes. They are using their admin\_name and password to login and manage the database.

The Volunteer entity stores detailed information about volunteers, such as their first\_name, last\_name, email, mobile, age, and address, comprising city and country. Volunteers are managed by admins and can have their history accessed when needed.

The Donation entity captures the details of donations, including donor information like first\_name, last\_name, email, and mobile. Additionally, it records donation\_date, amount, and card\_number for each donation, all uniquely identified by a donation\_id. This ensures that donation histories, including donor details, can be efficiently stored and retrieved for reporting or analysis. It allows admin to maintain a clear history of all donations, track donor contributions over time, and ensure transparency in financial reporting.

Similarly, the Contact entity is designed to store messages sent to the admins. Each message is linked to a contact\_id and includes the sender’s frist\_name, last\_name, email, and the message content, allowing admins to track and respond to communications effectively.

The relationship between Admin and other entities is defined by a "Manage" relationship, allowing admins to perform CRUD (Create, Read, Update, Delete) operations on volunteers, donations, and messages. This structure centralizes volunteer and donor data, facilitating streamlined management and reporting while maintaining communication records in the Contact entity. Overall, the database is organized to ensure seamless access to volunteer and donation histories as well as message management.

***Page description***

Home Page

At the top of the page, the navigation bar prominently displays the title as well as logo "Save Every Children" on the left side, styled in bold to emphasize the website's focus. This title also functions as a button, directing users back to the home page (the current page) when clicked. On the right side of the navigation bar, there are five clickable buttons labelled "Home", "Volunteer", "Contact", "Donation" and "Admin" that provide easy navigation to the corresponding sections of the site. These buttons are styled with white text, and upon hovering, the text color smoothly transitions from white to black with the background of white color, enhancing interactivity and user-friendliness.

To maintain continuity, the navigation bar is fixed on the top of the site with a background of light beige tone, ensuring ease of access regardless of where the user scrolls.

When scrolling down, it shows an impactful statement which is prominently displayed to immediately capture the user's attention and emphasize the urgency of the issue.

Below the statement, three icons representing Food, Water, and Medicine are displayed. These icons are designed to symbolize the core areas of support provided by the organization.

It also consists of a Donation Encouragement Section. This is designed to engage users both visually and emotionally, motivating them to contribute to the cause. On the left side, a large, high-quality image portrays children who are directly benefiting from the support provided by donations, creating a strong visual connection between the viewer and the impact of their contribution. This image serves as a powerful reminder of the positive difference their help can make. On the right side, compelling text emphasizes our organization is willing to help them, encouraging users to take action to support us. A prominent "Support Us" button is strategically placed to make it easy for users to contribute immediately.

The background of this section is dynamic, featuring colorful, animated shapes that move up and down as the user scrolls. This animation adds energy and draws attention to the section, making the call to action even more compelling and engaging.

Below the Donation Encouragement Section, the website highlights how users' support directly impacts children's lives. This section is designed with three distinct cards, each representing a specific area where donations can make a difference. The cards display key information about how different donation amounts contribute to the cause.

Volunteer Page

Volunteer page displays a volunteer application form, requires first name, last name, email, mobile, city and country. This allows user to submit their information to register as a volunteer. Thus, users interested in volunteering can fill out the registration form to join. Once submitted, the information is stored in the Volunteer entity for admin review and management.

Donation Page

Donation page is dedicated to accepting and managing financial contributions. It provides a donation form where users can input their donation amount and card details, as well as collects donor’s information, including first name, last name, email and mobile number. Thus, users can enter their information and make donations using the secure payment form. Then, donor information and transaction details are stored in the Donation entity for admin tracking.

Contact Page

Contact page is for users to send messages or inquiries to the organization. It displays a contact form for users to input their name, email, and message. It also provides the organization’s contact details for alternative communication. Thus, users can submit the contact form to send messages, which are stored in the Contact entity for admin review. Admins can thereby respond to inquiries via the email addresses provided.

Admin Page

The admin page is a secure and centralized interface that allows administrators to manage the entire system effectively. Accessible only via a login form requiring the registered and correct input of admin name and password. This page serves as a path to access admin dashboard.

Admin Dashboard

The Admin Dashboard serves as a centralized platform for managing administrative tasks and system features. It is designed to be intuitive, user-friendly, and responsive, catering to both desktop and mobile users. The layout consists of three primary sections: the header, the sidebar, and the main content area. The header, positioned at the top, includes page title "Admin Dashboard", and a convenient "Logout" button which is linked to the admin login page.

On the left side, the sidebar displays the "Welcome" text, admin icon and navigation links to add new admin in a column structure with a background color of dark blue.

The main content area, which occupies the central part of the layout, showcases three cards which represent Volunteer, Donation and Contact table stored in the database, allowing admin to do CRUD operations by clicking the "Manage" button.

Add New Admin page

The "Add New Admin" page features as a clean and straightforward form in the main content area that allows administrators to input an admin name, password, and confirm the password, ensuring data validation for secure account creation. A "Save" button, strategically placed at the bottom-right corner, provides quick submission functionality. There is also a "Back to Admin Dashboard" button on the sidebar on the left to let admin seamlessly navigate to admin dashboard page.   
   
Mobile interfaces for each page

Our websites design emphasizes responsiveness, ensuring smooth navigation on various devices. On desktop, most of the contents are displayed side by side, while on smaller screens, the layout adjusts seamlessly into column, making the contents display vertically for ensuring better experience to mobile users.