Charity Management System

***Team No:22***

# ***Abstract:***

The Charity Management System is a comprehensive software solution designed to streamline the operations and management processes of charitable organizations. This system provides a centralized platform for managing various aspects of charity work, including donation tracking, volunteer management, event coordination, fundraising activities, and beneficiary assistance. Through intuitive user interfaces, administrators can efficiently organize and monitor donation drives, allocate resources effectively, communicate with volunteers and donors, and generate insightful reports for performance evaluation and decision-making. Beneficiaries can also interact with the system to access support services and track their progress. By leveraging technology, the Charity Management System aims to enhance transparency, accountability, and efficiency in charitable endeavors, ultimately enabling organizations to better serve their communities and fulfill their missions.



# ***Introduction:***

The Charity Management System (CMS) is a sophisticated software platform developed to revolutionize the way charitable organizations operate. With a user-friendly interface and powerful backend capabilities, CMS empowers charities to efficiently manage their activities, from donor engagement to resource allocation. Leveraging Java programming language and database connectivity, CMS ensures robust performance and scalability. Through features like donation tracking, volunteer management, and event coordination, organizations can optimize their operations and maximize their impact on society. By harnessing the power of technology, CMS equips charities with the tools they need to thrive in the digital age and make a meaningful difference in communities worldwide.

Key Points about the Charity Management System:

* + **Donation Tracking**: CMS allows charities to track donations effectively, recording donor information, donation amounts, and donation types to facilitate transparency and accountability.
  + **Volunteer Management**: The system streamlines volunteer recruitment, scheduling, and communication, ensuring that charities can efficiently mobilize their volunteer workforce for various initiatives.
  + **Event Coordination**: CMS simplifies the process of organizing charity events, from planning and promotion to registration and attendee management, facilitating successful fundraising efforts.
  + **Fundraising Activities**: Charities can leverage CMS to launch and manage fundraising campaigns, track progress, and analyze results to optimize fundraising strategies.
  + **Beneficiary Assistance**: The system facilitates the delivery of support services to beneficiaries, enabling charities to track beneficiary information, needs, and progress effectively.
  + **Performance Evaluation**: CMS provides comprehensive reporting and analytics capabilities, allowing organizations to assess their performance, identify trends, and make data-driven decisions for continuous improvement.
  + **Security and Compliance**: With robust security features and adherence to data protection regulations, CMS ensures the confidentiality and integrity of sensitive information, fostering trust among donors, volunteers, and beneficiaries.
  + **Scalability and Customization:** Built using scalable architecture, CMS can accommodate the evolving needs of charities of all sizes, with the flexibility to customize features and workflows to suit specific organizational requirements.
  + .

**Functional Requirements:**

Functional Requirements define the fundamental actions or tasks that a system must perform. Here's a draft of some matter about the functional requirements for the Charity Management System:

**Functional Requirements for the Charity Management System:**

**1.User Authentication and Authorization**: The system should provide secure login functionality for administrators, volunteers, donors, and beneficiaries, with role-based access control to restrict unauthorized access to sensitive information.

**2.Donation Management**: Users should be able to record and track donations, including donor details, donation amounts, payment methods, and donation purposes, with the ability to generate donation receipts and acknowledgments.

**3.Volunteer Management**: The system should facilitate volunteer registration, scheduling, and communication, allowing administrators to assign tasks, track volunteer hours, and recognize volunteer contributions.

**4.Event Management**: Users should be able to create, manage, and promote charity events, including fundraisers, awareness campaigns, and community outreach programs, with features for event registration, attendance tracking, and feedback collection.

**5.Fundraising Campaigns:** The system should support the launch and management of fundraising campaigns, allowing users to set fundraising goals, track progress, and accept online donations through secure payment gateways.

**6.Beneficiary Assistance:** Users should be able to register beneficiaries, assess their needs, and provide support services such as food aid, shelter assistance, healthcare, and education, with features for case management and progress tracking.

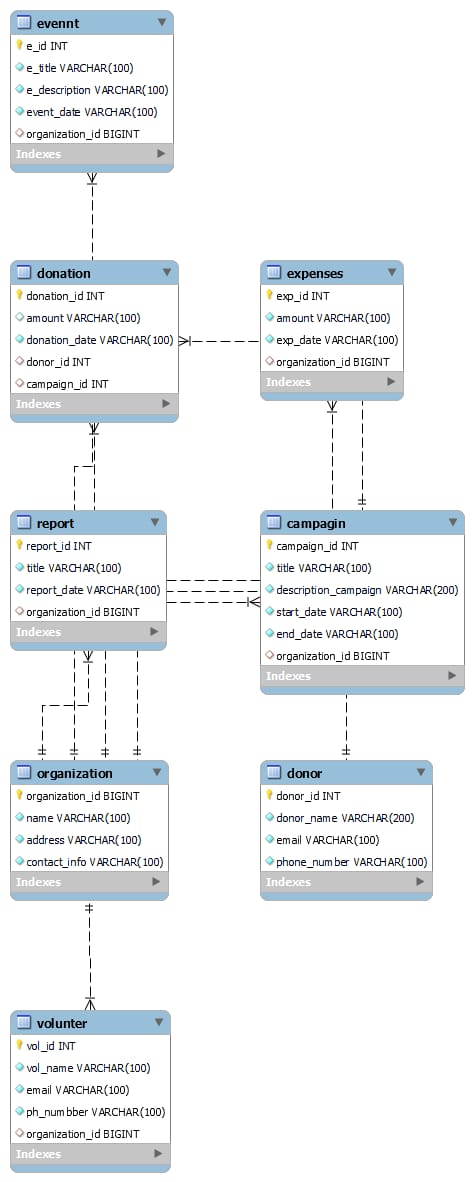
**7.Reporting and Analytics**: The system should offer comprehensive reporting and analytics tools, enabling users to generate various reports, dashboards, and metrics to evaluate performance, measure impact, and make data-driven decisions.

**8.Communication Tools**: The system should include communication features such as email notifications, newsletters, and messaging capabilities to engage with donors, volunteers, beneficiaries, and other stakeholders effectively.

**9.Integration with External Systems**: The system should allow integration with external systems and services, such as payment processors, CRM platforms, and social media platforms, to streamline data exchange and enhance functionality.

**10.Accessibility and Usability**: The system should be user-friendly, intuitive, and accessible to users of all abilities, with responsive design and support for multiple languages to accommodate diverse user demographics.

***ER Diagram:***



# **Quries to create Database:**

SHOW databases;

use charity\_management\_sysytem;

create table organization (

organization\_id bigint(100),name varchar(100) not null,

address varchar(100) not null, contact\_info varchar(100) not null,primary key (organization\_id));

create table donor(

donor\_id int(100),donor\_name varchar(200) not null, email varchar(100) not null, phone\_number varchar(100) not null,primary key (donor\_id));

create table campagin(

campaign\_id int(11),title varchar(100) not null, description\_campaign varchar(200) not null,start\_date varchar(100) not null, end\_date varchar(100) not null,organization\_id bigint(100),primary key (campaign\_id), foreign key (organization\_id) references organization (organization\_id));

create table donation(

donation\_id int(100),amount varchar(100),

donation\_date varchar(100) not null,

donor\_id int(100),

campaign\_id int(11),

primary key (donation\_id),

foreign key (donor\_id) references donor (donor\_id),

foreign key (campaign\_id) references campagin (campaign\_id));

create table volunter(

vol\_id int(100),vol\_name varchar(100) not null,email varchar(100) not null,ph\_numbber varchar(100) not null,organization\_id bigint(100),primary key (vol\_id), foreign key (organization\_id) references organization (organization\_id));

create table Evennt(

e\_id int(100),e\_title varchar(100) not null,e\_description varchar(100) not null, event\_date varchar(100) not null,organization\_id bigint(100),primary key (e\_id), foreign key (organization\_id) references organization (organization\_id));

create table expenses(

exp\_id int(100), amount varchar(100) not null,exp\_date varchar(100) not null, organization\_id bigint(100),primary key (exp\_id),foreign key (organization\_id) references organization (organization\_id));

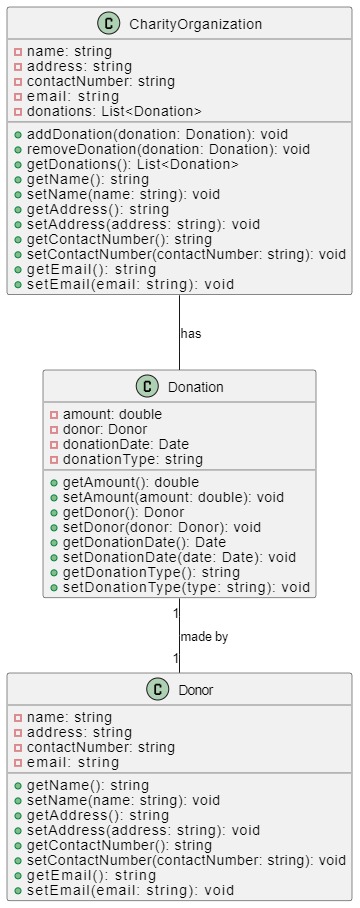
create table report(

report\_id int(100),title varchar(100) not null,report\_date varchar(100) not null,

organization\_id bigint(100),primary key (report\_id),

foreign key (organization\_id) references organization (organization\_id));

**UML Diagram of Student Information:**



**ALL JAVA FILES CODES:**

# ***Challenges List:***

# challenges that might arise specifically during the development of the Charity Management System project:

# **Database Design**: Designing a database schema that accommodates complex relationships between entities such as donors, campaigns, volunteers, events, and expenses, while maintaining data integrity and performance.

# **Integration Challenges**: Integrating various modules and components of the system, such as donation tracking, volunteer management, event coordination, and reporting, to ensure seamless data flow and interoperability.

# **Security** : Implementing security measures to protect sensitive data, prevent unauthorized access, and mitigate risks such as SQL injection, cross-site scripting (XSS), and data leakage.

# **Performance Optimization**: Optimizing system performance to minimize latency, improve response times, and handle concurrent user interactions efficiently, especially during peak usage periods.

# **Project Management**: Managing project timelines, milestones, and deliverables effectively to ensure on-time and within-budget completion of the Charity Management System while balancing competing priorities and resource constraints.

# 