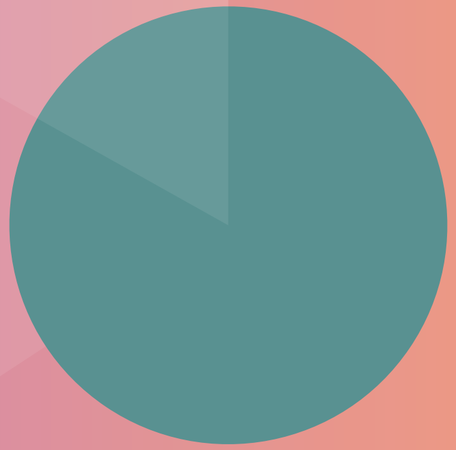
NGO MANAGEMENT SYSTEM Presentation



Dadi Sasank Kumar(22CS10020)
Nerella Trilochan(22CS10048)
Meda Abhinav(22CS10043)

Problem Description

The NGO dedicated to providing free school education for underprivileged children faces several critical management challenges. These include maintaining detailed records of registered students, tracking their educational progress and financial needs, estimating required funds for various educational expenses, prioritizing aid based on academic performance and socio-economic factors, managing donor pledges and contributions, tracking non-monetary donations such as books and clothing, and meticulously recording all expenditures. An efficient system is crucial to ensure effective allocation of resources and sustained support for the organization's mission.

Technical Details for NGO Management System

The NGO Management System is a web-based application built using Python Flask for the backend and Bootstrap, HTML, CSS, and JavaScript for the frontend. It serves as a comprehensive platform for managing various aspects of an NGO including student registration, donor management, volunteer registration, and communication.

Technologies Used

- **Backend**: Python Flask
- **Frontend**: Bootstrap, HTML, CSS, JavaScript
- **Database**: MySQL
- **Third-party Services**: FormCarry for contact form mailing service
- **Security**: Hardcoded login credentials, hCaptcha verification for login form

Features

1. **Admin Dashboard**: Accessible only after logging in, provides a centralized interface for managing all aspects of the NGO.
2. **Student Registration**:
    - Admin can register new students for the NGO.
    - Update and delete existing student details.
    - Approve voluntary student registrations.
3. **Donor Management**:
    - Keep track of donors.
    - Allow donors to donate through the website.
4. **Volunteer Registration**:
    - Volunteers can register on the website.
    - Volunteers can update their details after login.
5. **Authentication and Security**:
    - Hardcoded login credentials for admin access.
    - hCaptcha verification for added security on the login form.
6. **Contact Us Feature**:
    - Utilizes a third-party mailing service called "FormCarry" to directly mail the admin from the website's contact form.

Technical Details for NGO Management System

Architecture

1. **Frontend**:
    - Utilizes Bootstrap for responsive design and layout.
    - HTML templates are rendered dynamically using Flask's templating engine.
    - CSS and JavaScript for styling and interactivity.
2. **Backend**:
    - Flask handles routing, request handling, and response rendering.
    - Uses Flask forms for form validation and data handling.
    - SQLite database for storing NGO-related data.
3. **Third-party Integration**:
    - Integrates with FormCarry API for contact form mailing functionality.

Use Cases

- **Admin Registration and Login**:
  - Registration and login are facilitated for the admin of the NGO using hardcoded credentials and hCaptcha verification for security.
- **Admin Dashboard Access**:
  - Upon successful login, the admin is directed to the admin dashboard where various functionalities can be accessed.
- **Student Management**:
  - New students can be registered for the NGO, and existing student details can be updated and deleted as necessary by the admin.
  - Voluntary student registrations are approved by the admin.
- **Donor Management**:
  - Donors and their donation history are tracked by the admin.
  - Donations can be made to the NGO through the website by donors.
- **Volunteer Registration and Management**:
  - Volunteers can register on the website by providing their details.
  - Volunteer details can be updated after logging in.
- **Contact Us Feature**:
  - Users can contact the admin of the NGO by filling out a contact form on the website.
  - Messages from the contact form are received by the admin directly into their inbox using the FormCarry mailing service.
- **Security Measures**:
  - hCaptcha verification is utilized for added security during the login process.
  - Secure password policies are enforced, and sensitive data is protected by the system.
- **Error Handling**:
  - Clear error messages are provided, and unexpected errors are handled gracefully by the system.
- **Accessibility**:
  - The website is accessible on different devices and screen sizes.
  - Navigation and usability of the website are designed to be user-friendly.
- **Data Protection and Privacy**:
  - User data is securely stored and handled according to privacy regulations.

Possible Improvements

- **Enhanced User Interface**: Improve the overall user interface and experience by incorporating modern design principles, intuitive navigation, and responsive layouts to ensure accessibility across different devices.
- **OTP Authentication**: Consider implementing OTP (One-Time Password) authentication to enhance the security of user accounts. This would involve integrating OTP generation and verification mechanisms into the login process.
- **Automated Testing**: Introduce automated testing methodologies such as unit tests and integration tests to ensure the reliability and stability of the application. This would help identify and fix bugs more efficiently during development and future updates.
- **Automated Email Notifications**: Implement a feature to automatically send email notifications to students who register voluntarily on the website after their approval by the admin. Upon approval, students receive an email confirming that their application has been approved, along with any relevant information or instructions regarding their enrollment in the NGO program. This automated communication streamlines the onboarding process for students and ensures timely and consistent notifications, enhancing their overall experience with the NGO Management System.
- **Tabular Interfaces for Admin**: Implement tabular interfaces within the admin dashboard to provide organized and easily navigable views of key data such as donations, students, volunteers, and other relevant information. These tabular interfaces can include features such as sorting, filtering, and pagination to facilitate efficient data management and analysis. Admins can view detailed information, perform actions (such as editing or deleting records), and generate reports directly from these tabular interfaces, enhancing their ability to track and manage various aspects of the NGO's operations effectively.

Challenges faced

- Depending on the complexity of the website, there may be a need to acquire new technical skills or utilize additional tools. This could involve learning web development languages such as HTML, CSS, and JavaScript, as well as exploring frameworks like Bootstrap or React.
- The system is designed to meet the needs of various users, including admins, students, volunteers, donors, and beneficiaries. Ensuring that the website is user-friendly for each of these groups presents a significant challenge.
- For instance, when considering implementing pop-up boxes, we initially lacked the expertise and turned to JavaScript to address this requirement.
- Linking the frontend and backend of our website posed challenges that we had to overcome during the development process.
- We didn't incorporate a real payment gateway due to the complexity involved. Instead, we opted for an alternative solution to avoid complications.
- Similarly, we didn't integrate OTP authentication as it would have required significant modifications to the codebase.

Thankyou

Team Debuggers