**ABSTRACT**

The Education Donation Website aims to connect donors with financially disadvantaged students, providing a platform to support educational aspirations. Students who are unable to afford tuition fees, books, or other academic essentials can apply for financial aid. Donors can contribute to specific cases or a general fund while ensuring transparency and accountability. Built with a user-friendly interface, secure transactions, and efficient data handling, the platform bridges the gap between need and opportunity, fostering equitable access to education.

It is developed by using HTML, CSS, JAVASCRIPT, BOOTSTRAP as frontend and DJANGO, MYSQL as backend.

**TABLE OF CONTENT**

**INDEX**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **DESCRIPTION** | **PAGE. NO** |
| 1 | **INTRODUCTION** | 1 |
| 2 | **SYSTEM STUDY**  2.1 Existing System  2.2 Proposed System  2.3 Problem Definition and Project Description | 2  3 |
| 3 | **SYSTEM ANALYSIS**  3.1 Requirements specification  3.2 Feasibility Study | 4  5 |
| 4 | **SYSTEM DESIGN**  4.1 Architectural Design  4.2 Data Flow Diagram  4.3 Data Dictionary  4.4 User Interface Design  4.5 Normalization | 7  8  10  12  16 |
| 5 | **SYSTEM TESTING**  5.1 Types Of Testing  5.2 Types Of Validations  5.3 Error Message | 18  20  21 |
| 6 | **USER MANUAL**  6.1 Installation Manual  6.2 Operational Manual | 22  24 |
| 7 | **SYSTEM IMPLEMENTATION**  7.1Specail Features Of The Language | 24 |
| 8 | **FUTURE ENHANCEMENT** | 26 |
| 9 | **CONCLUSION** | 27 |
| 10 | **BIBLIOGRAPHY** | 28 |
| 11 | **APPENDIX**  11.1 Sample Screen Layout | 29 |

**INTRODUCTION**

**1.INTRODUCTION**

The Education Donation Website is a transformative platform designed to bridge the gap between generous donors and students struggling with financial barriers to education. Recognizing education as a fundamental right, the platform empowers community-driven change by connecting donors directly with deserving students, ensuring every contribution creates a tangible, lasting impact.

With a focus on transparency, the platform carefully verifies student profiles, allowing donors to track how their funds are used—whether for tuition, textbooks, or essential tools like laptops. This openness builds trust and fosters a deeper emotional connection between donors and students.

The platform also emphasizes engagement, turning donations into meaningful relationships through updates, messages, and shared success stories. Enhanced security measures ensure donor confidence, safeguarding both financial transactions and personal data.

From covering tuition to providing essential resources, the platform helps level the educational playing field, opening doors to brighter futures for students and their communities. More than just a website, it’s a movement to eradicate educational inequality and create a world where every individual can reach their full potential.

**SYSTEM STUDY**

**2.SYSTEM STUDY**

**2.1 EXISTING SYSTEM**

Traditional education donation methods lack structure, transparency, and engagement. In many cases, donors are unaware of how their contributions are utilized, leading to skepticism and reduced participation.

****Challenges in Existing Systems:****

* Limited transparency in fund allocation
* Lack of accountability in fund usage.
* Minimal interaction between donors and beneficiaries.

**2.2 **PROPOSED SYSTEM****

The Education Donation Website overcomes these challenges by introducing:

* **User Profiles:** Students can create profiles showcasing their academic achievements and financial needs.
* **Donation Categories:** Specific categories such as tuition fees, books, and supplies.
* **Transparency and Tracking:** Donors can track fund allocation and utilization.

**2.3 PROBLEM DEFINITION AND PROJECT DESCRIPTION**

Financial difficulties often hinder students from pursuing education, despite their potential. The project aims to build a platform where donors can provide financial support, ensuring transparency and accountability. By facilitating connections between donors and recipients, the platform empowers underprivileged students to continue their education. This software has different modules to ensure this problems they are

* Admin
* User

**ADMIN MODULE**

* View Fund Allocation
* View Donor List
* View Student List

**USER MODULE**

* Registration
* Uploading Documents
* Donating Fund
* Posting Feedback

**SYSTEM ANALYSIS**

**3.SYSTEM ANALYSIS**

**3.1 REQUIREMENTS SPECIFICTAION**

**HARDWARE REQUIREMENTS**

Processor : Intel

RAM : 4.00GB

Hard disk : 40GB

Monitor : 15”mVGA monitor

Keyboard : 104 keys keyboard

Mouse : Optical mouse

**SOFTWARE REQUIREMENTS**

Opersating system : 64-bit

Local host server : XamppServer

Database connectivity : phpMyAdmin

Front end : HTML, CSS, JavaScript and Bootstrap

Back end : Django and MYSQL

**3.2 FEASIBLITY STUDY**

The Education Donate Website is a simple and easy-to-use platform designed specifically for college students to support educational causes. It helps students raise funds for projects like scholarships, college events, or educational resources. This study checks if the project is practical and can work in the long run by looking at technical, operational, economic.

**TECHNICAL FEASIBILITY**

This checks if the technology needed to build and run the Education Donate Website is available.

Technology Used:

Frontend: HTML, CSS, JavaScript (to create a simple and user-friendly interface).

Backend: Django (a reliable framework for handling website functions).

Database: MySQL (to store user and fundraiser data securely). .

**OPERATIONAL FEASIBILITY**

This checks if college students can easily use the website.

Features for Students:

- Simple and clean design for easy navigation.

- Students can create fundraisers for their college projects or events.

- Real-time updates on fundraiser progress.

- Secure and quick donations.

Focus on College Students:

- Only college students can create fundraisers for educational needs.

- Donors (like friends, family, or alumni) can easily contribute.

- Students can track how much money has been raised and share updates.

**ECONOMIC FEABILITY**

This checks if the website can make enough money to cover its costs.

How It Will Earn Money:

- A small fee is charged on each donation.

- Colleges or student groups can pay for extra features like custom branding.

- Partnerships with educational organizations or sponsors.

Costs to Consider:

- Building and maintaining the website.

- Server costs for hosting the website.

- Marketing to attract students and donors.

**SYSTEM DESIGN**

**4**. **SYSTEM DESIGN**

**4.1 ARCHITECURAL DESIGN**

DONATION WEBSITE

ADMIN

USER

registration

Student request

Donating

Posting feedback

View fund allocation

View donor list

View student list

**4.2 DATA FLOW DIAGRAM**

**DFD level 0**

DATABASE

ADMIN

DONATION WEBSITE

USER

**DFD Level 1**

View fund Allocation

fundtb

ADMIN

studenttb

donortb

View Donor List

View Student List

registeration

logintb

Student request

documenttb

**USER**

donating

donatetb

Posting feedback

feedtb

**4.3 DATA DICTIONARY**

**Table Name:** donatetb

**Purpose:** to store the details of the donor’s

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1. No | Field Name | Data type | Size | Constraint | Description |
| 1 | first\_name | varchar | 50 | Not Null | Donor’s first name |
| 2 | last\_name | varchar | 50 | Not Null | Last name |
| 3 | phone\_number | varchar | 15 | Not Null | Phone number |
| 4 | email\_id | varchar | 100 | Not Null | Email address |
| 5 | pancard\_no | varchar | 10 | Not Null | Pan number |
| 6 | aadhar\_no | int | 12 | Null | aadhar |
| 7 | address | text |  | Not Null | Full address |
| 8 | country | varchar | 50 | Not Null | country |
| 9 | state | varchar | 50 | Not Null | state |
| 10 | city | varchar | 50 | Not Null | city |
| 11 | pincode | number | 10 | Not Null | Postal code |
| 12 | donation\_amount | decimal | 10,2 | Not Null | Amount donated |
| 13 | payment\_method | ENUM |  | Not Null | Mode of transaction |
| 14 | transaction\_id | varchar | 50 | Not Null | Unique id for each fund |
| 15 | donation\_date | DATETIME |  | Not Null | Date and time |

**Table Name:** logintb

**Purpose:** to store the username and password of user

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| S.No | Field Name | Data Type | Size | Constraint | Description |
| 1 | user\_id | int | 15 | unique | User id |
| 2 | email\_id | varchar | 100 | Not Null | Email address |
| 3 | password | varchar | 20 | Not Null | Password |
| 4 | role | varchar | 10 | Not Null | Role of user |

**Table Name:** feedtb

**Purpose:** to store the feedback given by users

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| S.No | Field Name | Data Type | Size | Constraint | Description |
| 1 | user\_id | int | 15 | unique | User id |
| 2 | name | varchar | 50 | Not Null | Name |
| 3 | feedback | text |  | Not Null | Feedback |

**Table Name:** documenttb

**Purpose:** to store the feedback given by users

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| S.No | Field Name | Data Type | Size | Constraint | Description |
| 1 | student\_id | int | 15 | Not Null | Student id |
| 2 | purpose | varchar | 100 | Not Null | Purpose of fund |
| 3 | Amount\_req | int | 20 | Not Null | Fund required |
| 4 | status | varchar | 20 | Not Null | status |

**Table Name:**

**Purpose:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| S.No | Field Name | Data Type | Size | Constraint | Description |
| 1 | user\_id | int | 15 | unique | User id |
| 2 | email\_id | varchar | 100 | Not Null | Email address |
| 3 | password | varchar | 20 | Not Null | Password |
| 4 | role | varchar | 10 | Not Null | Role of user |

**4.4 USER INTERFACE DESIGN**

**USER SIGNUP**

**SIGNUP**

**FULL NAME:**

**EMAIL:**

**PASSWORD:**

**PHONE:**

**ADDRESS:**

**Already have a account? login**

SUBMIT

**ADMIN LOGIN**

ADMIN LOGIN

USERNAME:

PASSWORD:

LOGIN

**LOGIN PAGE**

**LOGIN**

**Email:**

**Password:**

**Don’t have a account? signup**

**LOGIN**

**HOME PAGE**

LOGIN

📕📸🕊🔏

HOME CONTATC MYACTIVITY ADMIN ABOUT DASHBOARD

CONTACT US FEEDBACK FOLLOW US

Phone:1234567890

Email:james@gmail.com 📕📸🕊🔏

Address:Tirunelveli

DONATE

EMPOWERING DREAMS THROUGH EDUCATION

**DONATION PAGE**

MM/DD/YY

SUBMIT

FRIST NAME: LAST NAME:

EMAIL: PHONE:

ADDRESS: AADHAAR NUMBER:

PAN NUMBER: CITY:

STATE: COUNTRY:

DONATION AMOUNT: DONATION DATE :

**ADMIN DASHBOARD**

APPROVED APPROVAL

0

REJECTED APPROVAL

0

RECENET DONATION

PENDING APPROVAL

0

TOTAL DONATION

00.00

**4.5 NORMALIZATION**

Normalization is the systematic process of organizing data in a database to minimize redundancy within a relation or set of relations. Its primary goal is to eliminate undesirable characteristics such as Insertion, Update, and Deletion Anomalies. Normalization involves dividing a larger table into smaller tables and establishing relationships between them. The normalization process, which involves defining data relationships in a structured manner, is crucial for maintaining data integrity and efficiency.

**First Normal Form (1NF)**

For a table to achieve First Normal Form, it must adhere to the following four rules:

* All attributes/columns should be single-valued (atomic).
* Values stored in a column should belong to the same domain.
* Every column in a table should have a unique name.
* The order in which data is stored should not impact the interpretation.

**Second Normal Form (2NF)**

To attain Second Normal Form, a table should meet the following criteria:

* It must already be in First Normal Form.
* No Partial Dependency should exist, meaning a non-prime attribute should not be functionally dependent on part of a candidate key.

**Third Normal Form (3NF)**

A table achieves Third Normal Form when it satisfies the following conditions,

* It is already in Second Normal Form.

And, it doesn't have Transitive Dependency.

**Boyce and Codd Normal Form (BCNF)**

## 

Boyce and Codd Normal Form is a higher version of the Third Normal form. This form deals with certain type of anomaly that is not handled by 3NF. A 3NF table which does not have multiple overlapping candidate keys is said to be in BCNF. For a table to be in BCNF, following conditions must be satisfied:

* R must be in 3rd Normal Form
* For each functional dependency ( X → Y ), X should be a super Key.

**Fourth Normal Form (4NF)**

The table is said to be in the Fourth Normal Form When,

* It is already in Boyce-Codd Normal Form.
* And, it doesn't have Multi-Valued Dependency.

**SYSTEM TESTING**

1. **SYSTEM TESTING**
   1. **TYPES OF TESTING**

Testing serves as a pivotal quality assurance technique throughout the software development lifecycle. Post-coding, software programs become available for execution, primarily for the purpose of testing. Its primary goal is not only to uncover errors introduced during coding but also to detect any oversights from earlier phases, be it in requirements, design, or coding. The fundamental types of testing include:

• Unit Testing

• Integration Testing

• Validation Testing

• Output Testing

• User Acceptance Testing

**UNIT TESTING**

Unit testing constitutes the initial phase of testing. It involves scrutinizing different modules against the specifications outlined during their design. This process verifies the code produced during the coding of individual program modules within an isolated environment. Unit testing primarily emphasizes assessing modules independently to pinpoint errors. In the Donation Website for Student Scholarships, unit testing ensures that individual components, such as the donation processing module, student application form, and payment gateway, work correctly before integration.

**INTEGRATION TESTING**

Following individual module testing, integration testing becomes imperative. This stage ensures that the modules function cohesively by testing them in conjunction with one another, thereby validating the correctness of their interfaces. For example, in the Donation Website, integration testing ensures that the student application

module properly connects with the donor module, the database, and the payment system without errors.

**VALIDATION TESTING**

Validation testing represents the final assurance phase, ensuring that the software aligns with all functional, behavioral, and performance requirements. The software is integrated and evaluated as a unified package. Successful validation occurs when the software operates in a manner consistent with user expectations. For instance, in the Donation Website, validation testing checks whether students can successfully submit applications, donors can contribute funds securely, and the admin can verify and process transactions correctly.

**OUTPUT TESTING**

Subsequent to validation testing, output testing becomes paramount. This phase evaluates the output generated by the proposed system, vital for its utility. Output is assessed in two formats: on-screen and in reports. For example, the system should correctly display donation confirmation messages to donors, show real-time funding status to students, and generate reports for admin tracking of donations and scholarships.

**USER ACCEPTANCE TESTING**

User acceptance testing stands as a crucial determinant of system success. Throughout development, this process continuously engages prospective system users to gauge their acceptance. Adjustments are made as necessary to ensure alignment with user expectations and preferences. In the Donation Website, this involves gathering feedback from students, donors, and admins to ensure the website is user-friendly, secure, and meets all functional requirements before deployment.

* 1. **TYPES OF VALIDATIONS**

Validation controls are instrumental in ensuring the accuracy, security, and completeness of user input in the Donation Website for Student Scholarships. Various types of validations help prevent incorrect data entry and ensure the system functions as intended. The key validation types implemented in this project include:

**REQUIRED FIELD VALIDATIONS**

The Required Field Validator ensures that users input data into designated fields, preventing submissions with empty fields. This is essential for enforcing form completion and ensuring all necessary details are provided. For example, in the Donation Website, students must provide their name, email, and financial documents before submitting a scholarship application. Similarly, donors must enter valid payment details before making a contribution.

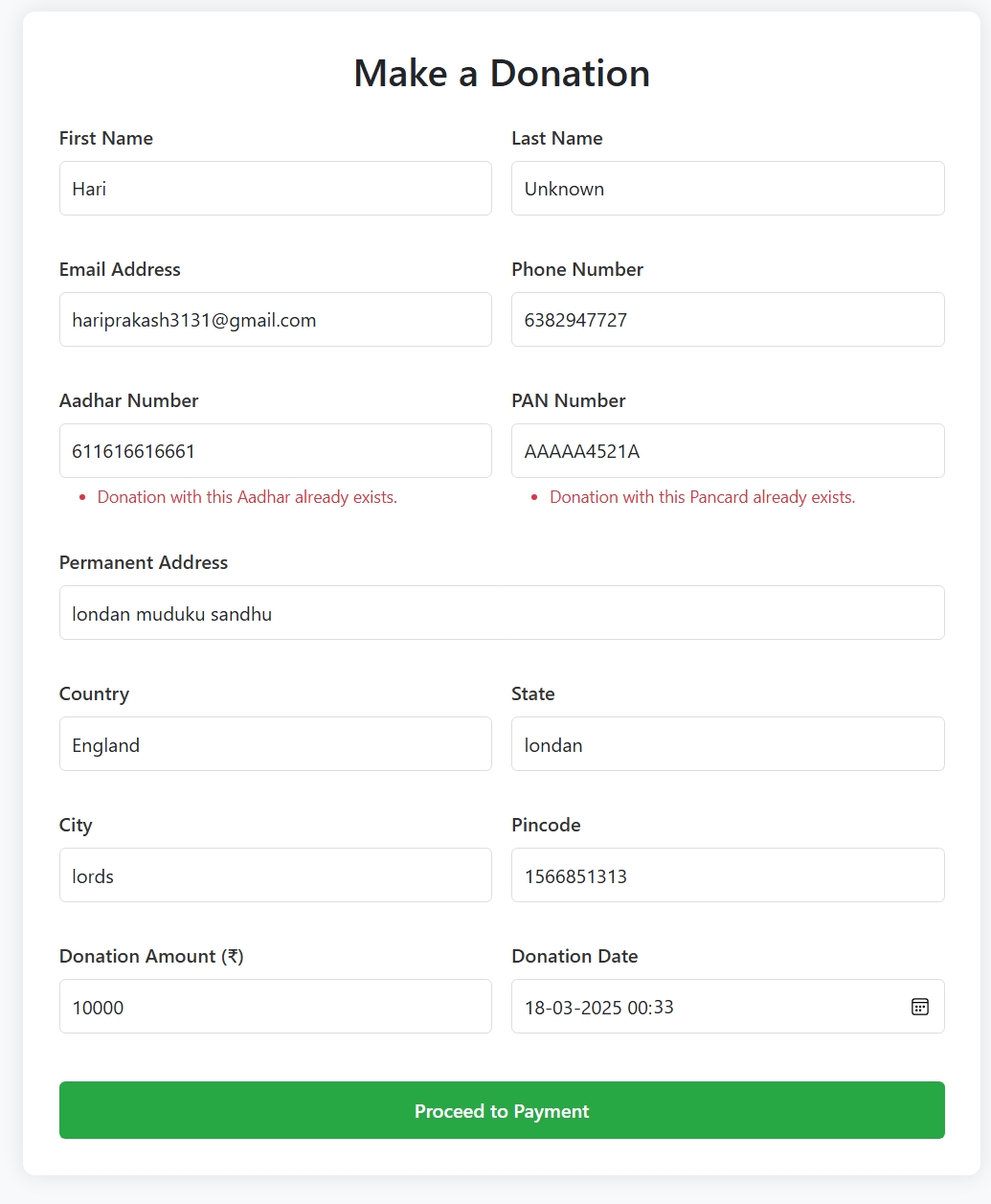
**REGULAR EXPRESSION VALIDATION**

The Regular Expression Validator is a versatile tool capable of validating a wide range of string formats. It employs regular expressions to verify user input against predefined patterns. In this project, it is used to validate email addresses, phone numbers, and password strength. For example, donor emails must follow the correct format (e.g., user@example.com), and student passwords must meet security requirements, including a mix of uppercase, lowercase, and special characters.

**RANGE VALIDATION**

The Range Validator ensures that user input falls within predefined acceptable values. This validation extends beyond numeric inputs to include date and text-based constraints. For instance, in the Donation Website, the system enforces valid phone number formats and ensures that donation amounts fall within an acceptable range (e.g., $10 - $10,000). Similarly, date fields such as scholarship deadlines are validated to prevent users from entering past dates.

**5.3 ERROR MESSAGES**

****

**USER MANUAL**

1. **USER MANUAL**
   1. **INSTALLATION MANUAL**

Follow these steps to install and set up the Donation Website for Student Scholarships using Django, MySQL Workbench, and HTML/CSS/Bootstrap:

• Download and install Python (latest stable version).

• Install Django framework using the command: `pip install django`.

• Install MySQL Workbench for database management.

• Install necessary database connectors with: `pip install mysqlclient`.

• Clone or download the project files to your local system.

• Open the terminal or command prompt and navigate to the project folder.

• Create and activate a virtual environment using: `python -m venv env` and `source env/bin/activate` (Linux/Mac) or `env\Scripts\activate` (Windows).

• Install required dependencies with: `pip install -r requirements.txt`.

• Configure the MySQL database settings in the `settings.py` file.

• Run database migrations using: `python manage.py makemigrations` and `python manage.py migrate`.

• Create a superuser for the admin panel using: `python manage.py createsuperuser`.

• Start the Django development server with: `python manage.py runserver`.

• Open a web browser and enter `http://127.0.0.1:8000/` to access the website.

• If the homepage loads correctly, the installation is successful, and the system is ready to use.

* 1. **OPERATIONAL MANUAL**

This section provides step-by-step instructions on how to use the Donation Website for Student Scholarships, including student applications, donor contributions, and admin functionalities.

**Admin Operations**

• Open the project and navigate to the home page.

• Log in to the admin panel using your credentials (`http://127.0.0.1:8000/admin/`).

• Upon successful login, you will be directed to the admin dashboard.

• View and manage registered users, including students and donors.

• Approve or reject student scholarship applications from the admin dashboard.

• View donation records and track the total amount donated.

• Manage system settings such as user roles and application status updates.

• Log out securely after completing administrative tasks.

**Student Operations**

• Go to the home page and click on 'Sign Up' to create an account.

• Log in using your registered email and password.

• Fill out the scholarship application form, including personal, family, and educational details

• Upload required documents such as Aadhar, PAN, and bank passbook.

• Submit your application and wait for approval from the admin.

• Track your application status from the dashboard.

• Log out securely after checking your application status.

**Donor Operations**

• Go to the home page and click on 'Donate Now'.

• Fill in your details, including name, email, and payment information.

• Select a student scholarship application to contribute towards.

• Complete the payment using the integrated payment gateway.

• Receive a confirmation message after a successful donation.

• Track your past donations in the dashboard if logged in.

• Log out securely after completing transactions.

**SYSTEM IMPLEMENTATION**

1. **SYSTEM IMPLEMENTATION**
   1. **SPECIAL FEATURES OF THE TECHNOLOGIES USED**

The Donation Website for Student Scholarships is built using Django, MySQL Workbench, HTML, CSS, Bootstrap, and JavaScript. Each technology offers unique features that contribute to the system's efficiency, security, and scalability.

**Django**

Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design. It was created to simplify complex web application development while ensuring security, scalability, and maintainability.

• MVC-Based Architecture: Django follows the Model-View-Controller (MVC) pattern, ensuring better separation of concerns.

• Built-in Authentication: It provides built-in user authentication and authorization management.

• Security Features: Django protects against SQL injection, XSS, CSRF, and other security threats by default.

• ORM (Object-Relational Mapping): Allows seamless interaction with the database using Python models instead of raw SQL queries.

• Scalability: Django can handle large-scale applications due to its modular and reusable architecture.

• Template System: Django’s templating engine helps separate UI design from application logic, ensuring clean and maintainable code.

**MySQL Workbench**

MySQL Workbench is a visual database design tool that provides a comprehensive interface for managing MySQL databases efficiently.

• Ease of Use: MySQL Workbench offers an intuitive interface for database design, development, and management.

• Robust Security: Supports encrypted connections, user role management, and access control for data security.

• Data Modeling: Provides powerful tools for designing and visualizing database schemas.

• Performance Tuning: Built-in tools help optimize database performance by analyzing queries and indexing.

• Cross-Platform Compatibility: MySQL Workbench runs on Windows, macOS, and Linux.

**HTML, CSS, and Bootstrap**

HTML, CSS, and Bootstrap are used to design the front-end interface of the donation website, ensuring responsiveness and user-friendly navigation.

• HTML5: Provides a structured way to define the website layout with semantic elements.

• CSS3: Enables styling, animations, and layout control for an enhanced user experience.

• Bootstrap Framework: Ensures mobile responsiveness with pre-designed components and a grid system.

• Cross-Browser Compatibility: Ensures consistent appearance across different web browsers.

**JavaScript**

JavaScript is used to enhance the interactivity and dynamic functionality of the donation website.

• Client-Side Scripting: Enables dynamic content updates without refreshing the page.

• Form Validation: Validates user inputs in real-time before submission.

• AJAX Integration: Allows asynchronous data fetching for better user experience.

• Event Handling: Captures user interactions like clicks and key presses for responsive behavior.

**FUTURE ENHANCEMENT**

1. **FUTURE ENHANCEMENTS**

The Donation Website for Student Scholarships can be further enhanced with additional features and improvements to enhance user experience, security, and efficiency. Some potential future enhancements include:

•**Razorpay Payment Gateway Integration**: Implementing Razorpay as an additional payment gateway will offer more secure and seamless transactions, supporting multiple payment modes including UPI, credit/debit cards, and net banking.

•**Mobile Application Development**: A mobile application for both Android and iOS users will allow students and donors to easily apply for scholarships and make donations on the go.

•**Automated Donation Receipts**: Enhance the system to automatically generate and email donation receipts to donors for better transparency and record-keeping.

•**AI-Based Scholarship Matching**: Implement artificial intelligence algorithms to match students with potential donors based on their financial needs and academic performance.

•**Multi-Currency and International Payments**: Enable donations from international donors by integrating multi-currency support and global payment processing options.

• **Blockchain for Transparency**: Using blockchain technology for tracking donations will ensure greater transparency and security in fund distribution.

•**SMS and Email Notifications**: Improve communication by sending automated SMS and email notifications for scholarship approvals, donation confirmations, and application updates.

•**Improved Admin Dashboard**: Enhance the admin panel with advanced analytics, real-time donation tracking, and graphical reports to better manage funds and applications.

**CONCLUSION**

1. **CONCLUSION**

The integration of technology and the widespread use of the internet have significantly enhanced the accessibility and efficiency of financial aid distribution. The Donation Website for Student Scholarships enables students to apply for scholarships and donors to contribute seamlessly, ensuring that financial constraints do not hinder education. Compared to traditional donation methods, this web-based system offers substantial benefits in terms of transparency, security, and ease of use.

Robust validation techniques have been implemented to ensure accurate data entry for donations and scholarship applications, facilitating a seamless and secure process. The system efficiently tracks donations, scholarship applications, and approvals, providing valuable insights into fund distribution. With an intuitive user interface, it ensures a user-friendly experience for students, donors, and administrators. Thorough testing has been conducted to ensure optimal performance, and any identified errors have been promptly addressed. The project meets user requirements, offering a reliable and secure platform for financial assistance. Looking ahead, there is potential for further enhancements, including the integration of Razorpay for a more seamless payment experience, AI-based scholarship matching, and blockchain technology for improved transparency. By incorporating these features, the system can continue to evolve and better serve its users in the future.

**BIBLIOGRAPHY**

1. **BIBLIOGRAPHY**

**AI TOOLS REFERENCES**

* Chatgpt
* Deepseek
* Windsurf
* trae

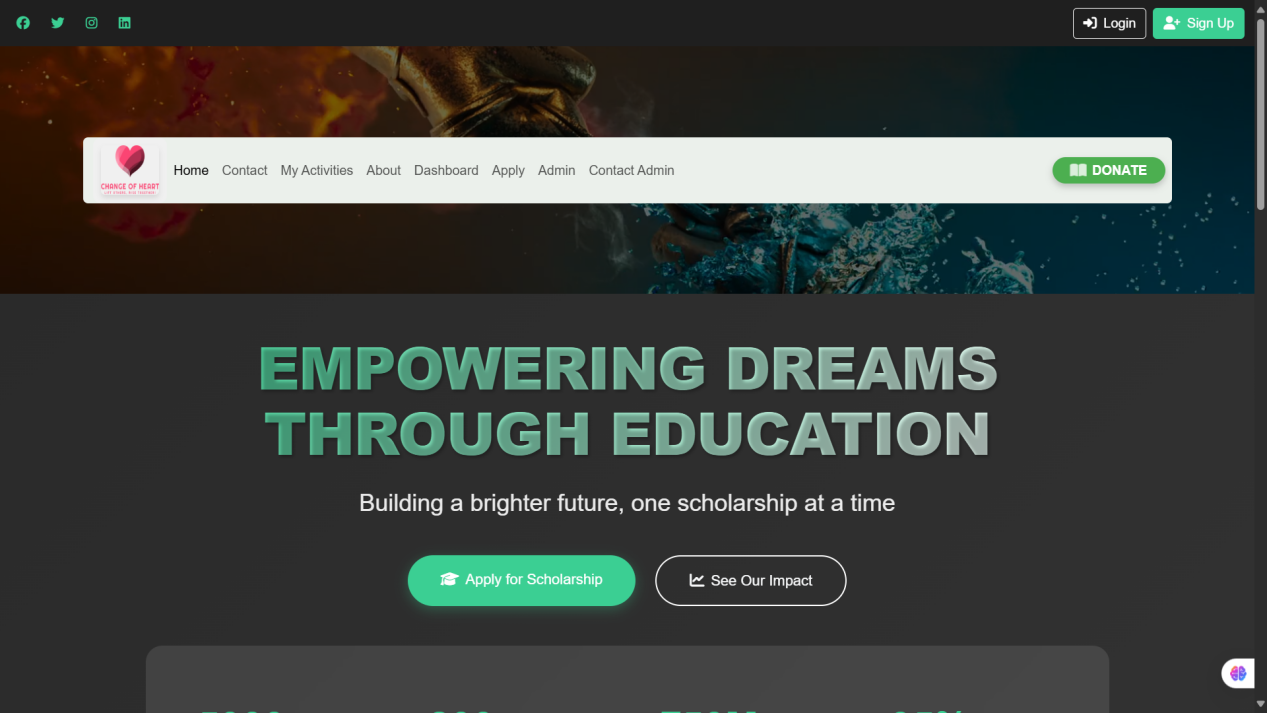
**WEBSITE REFERENCES**

* www. w3schools.com

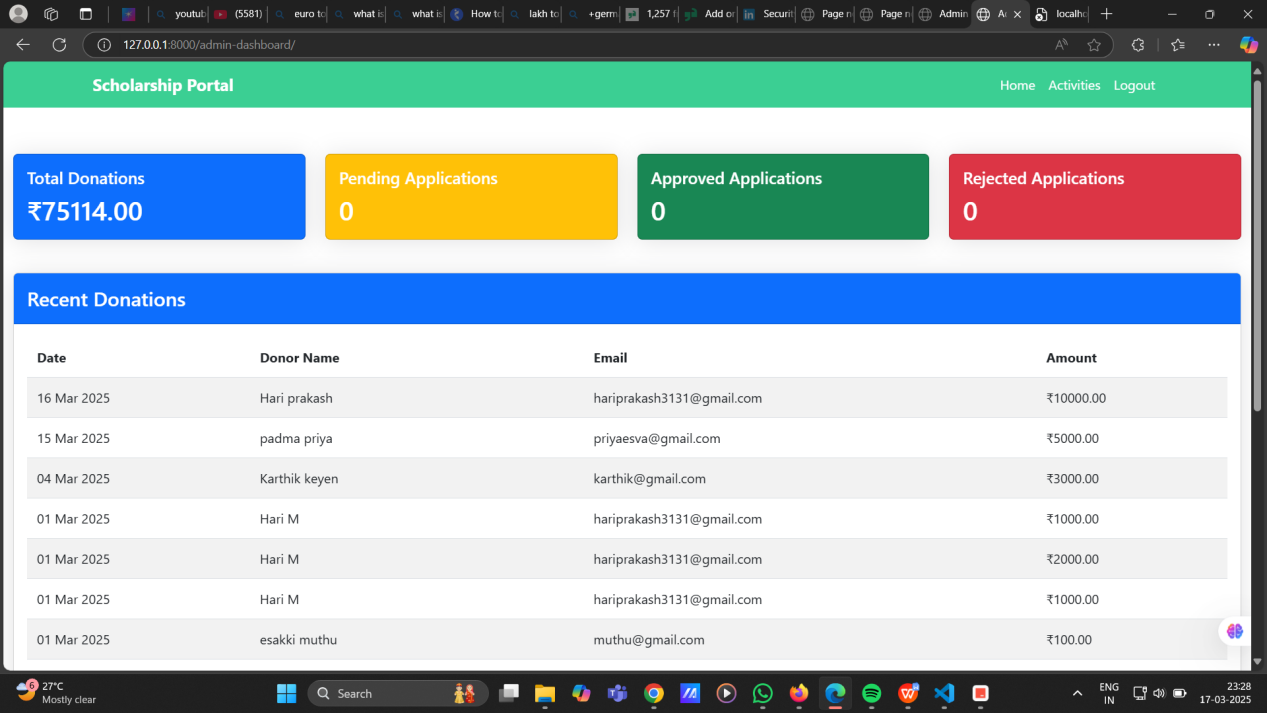
**APPENDIX**

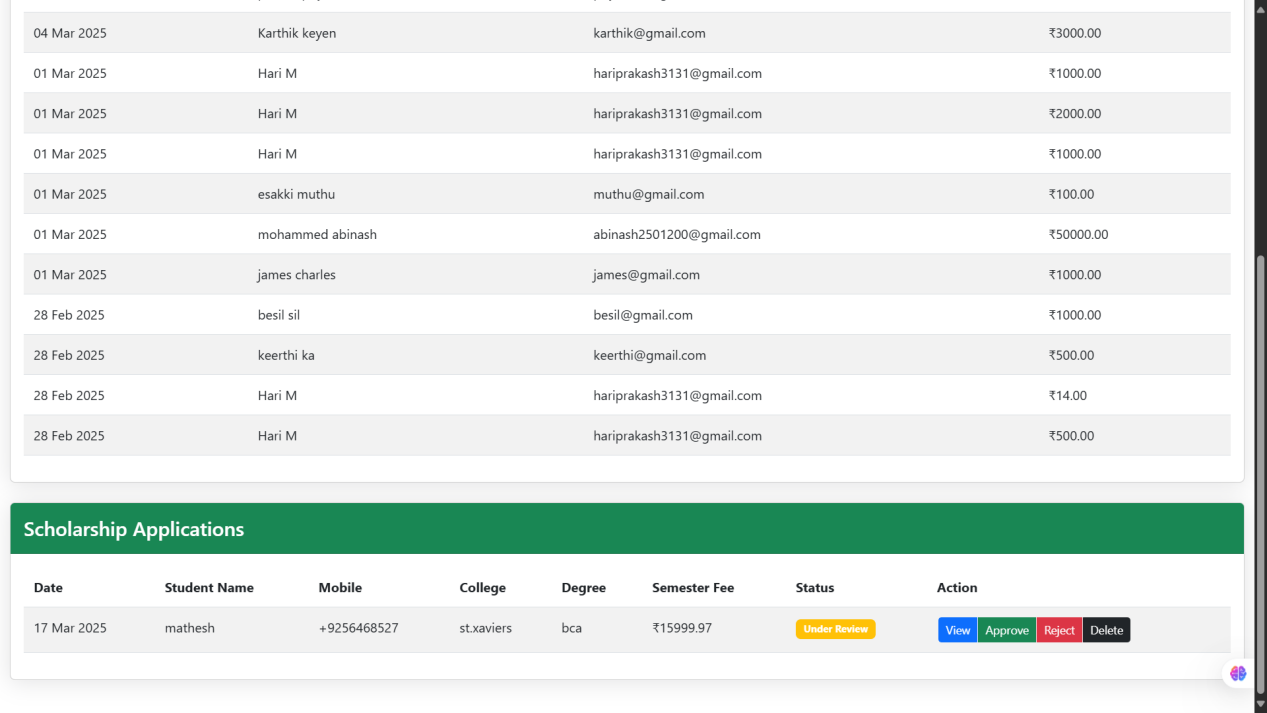
1. **APPENDIX**
2. **SAMPLE SCREEN LAYOUTS**

**Home page**

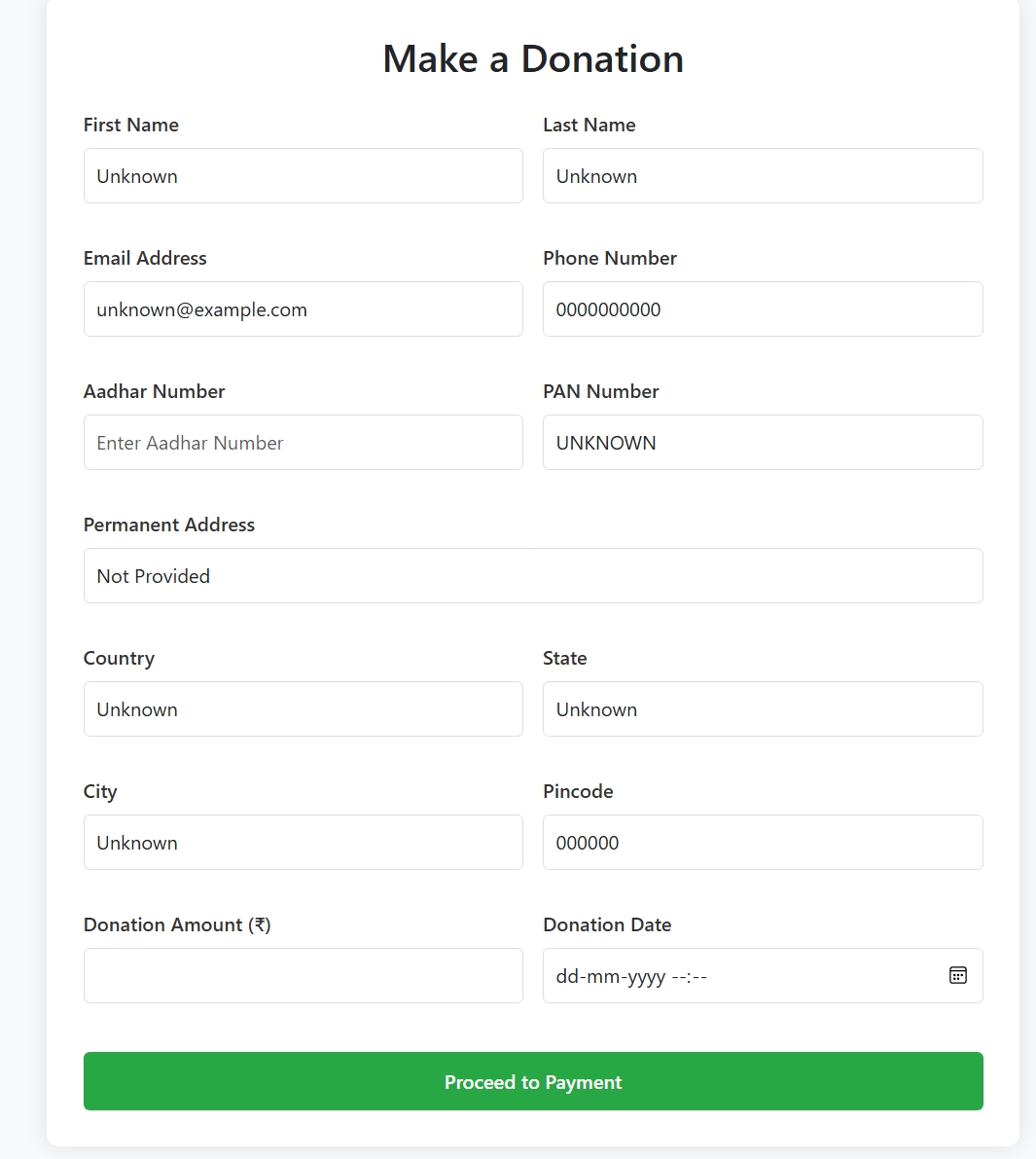
****

**Admin dashboard**

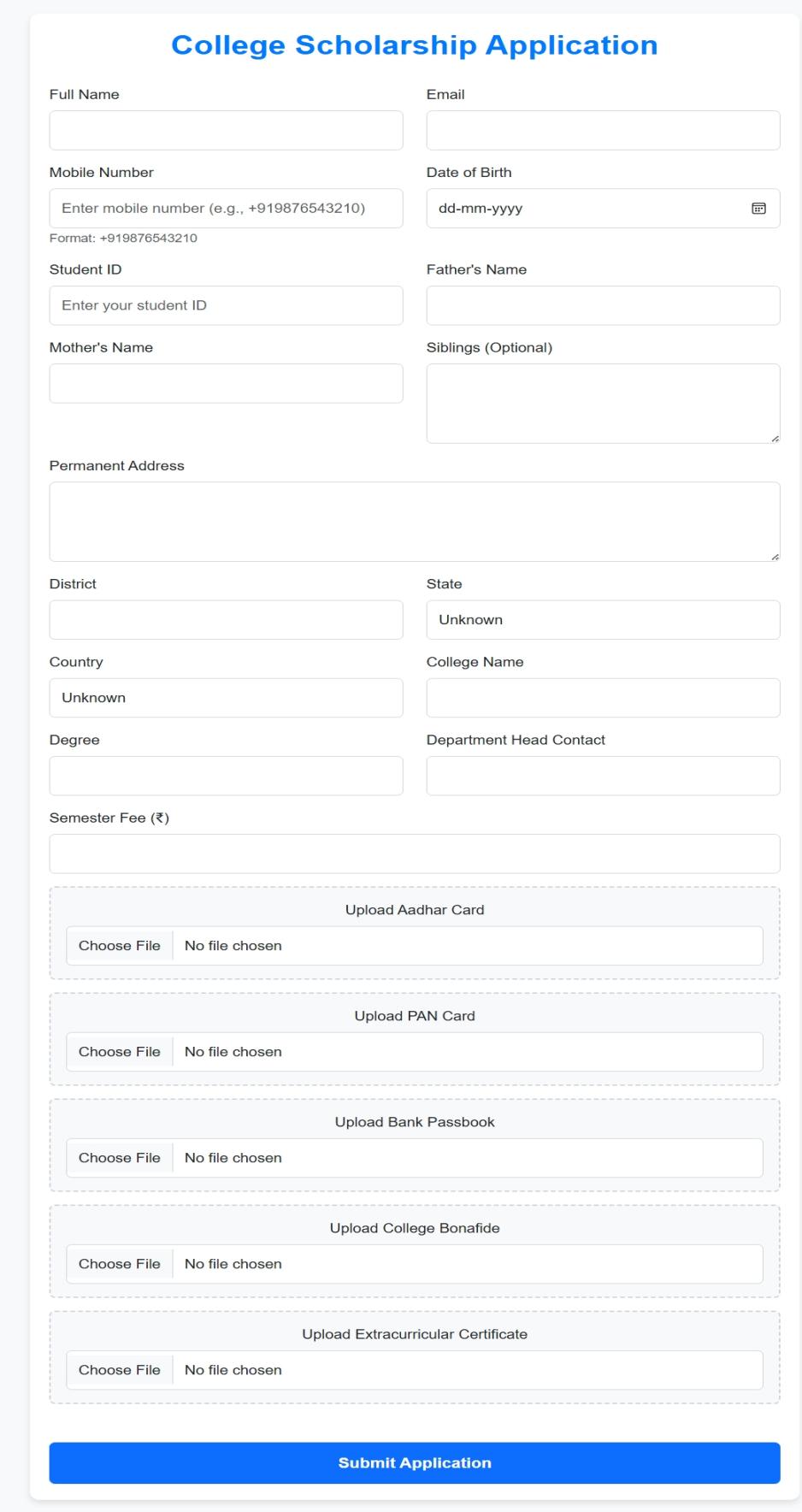
****

****

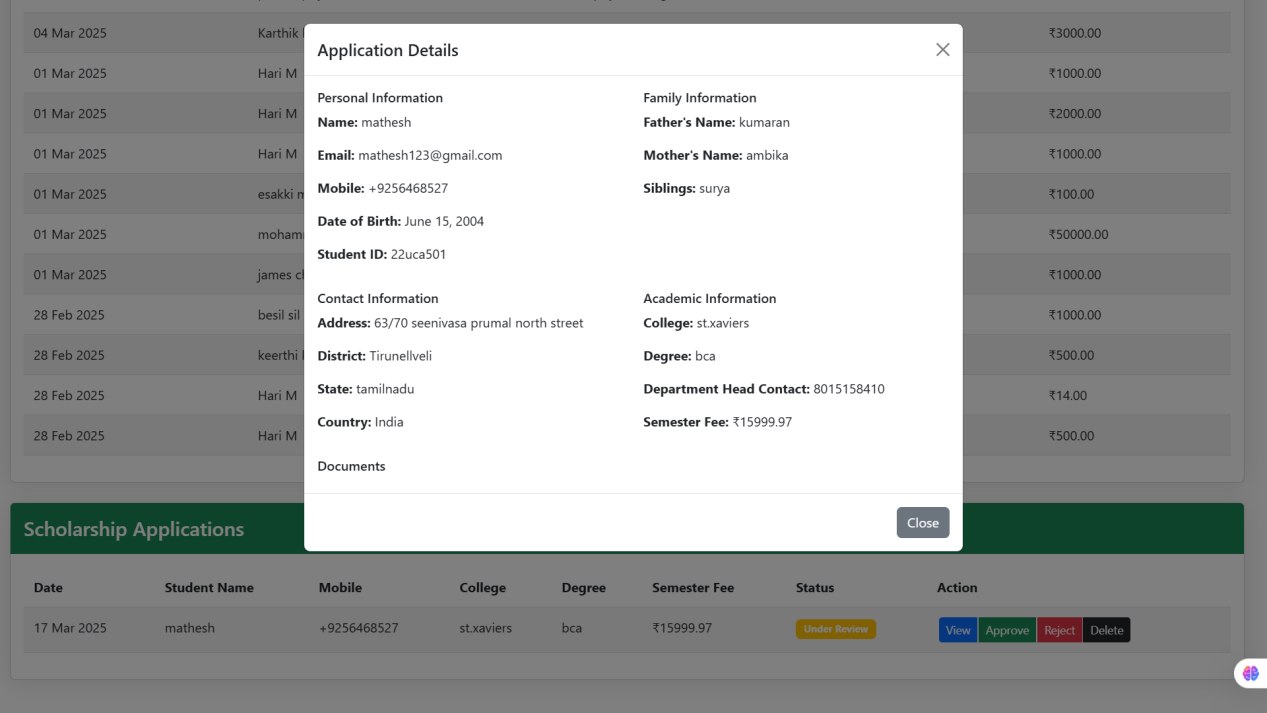
**Donation page**

****

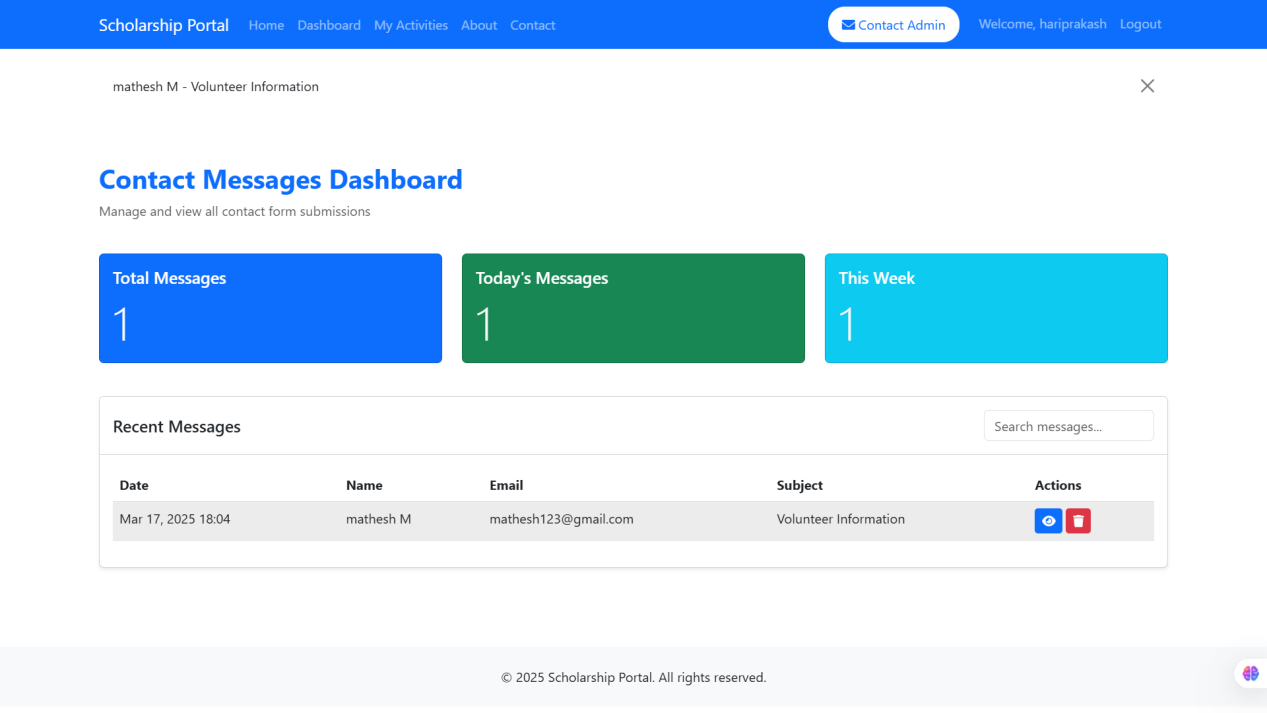
**Student application form**

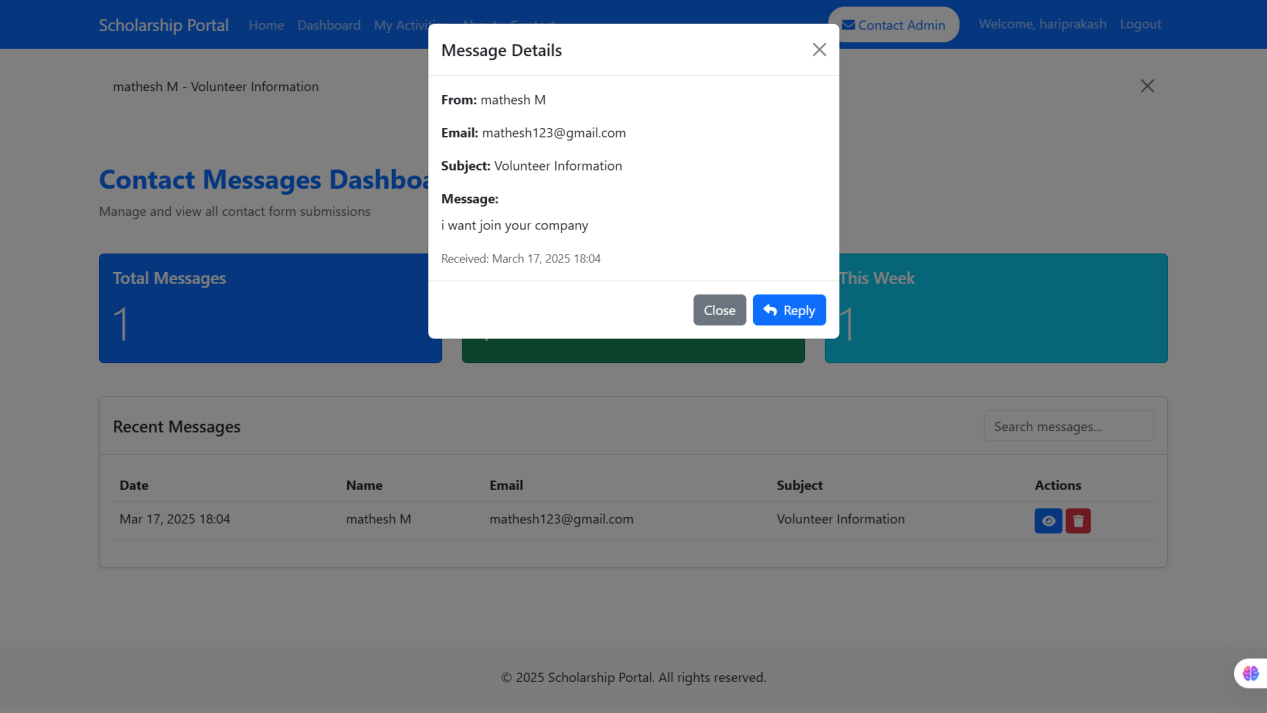
****

**Application details**

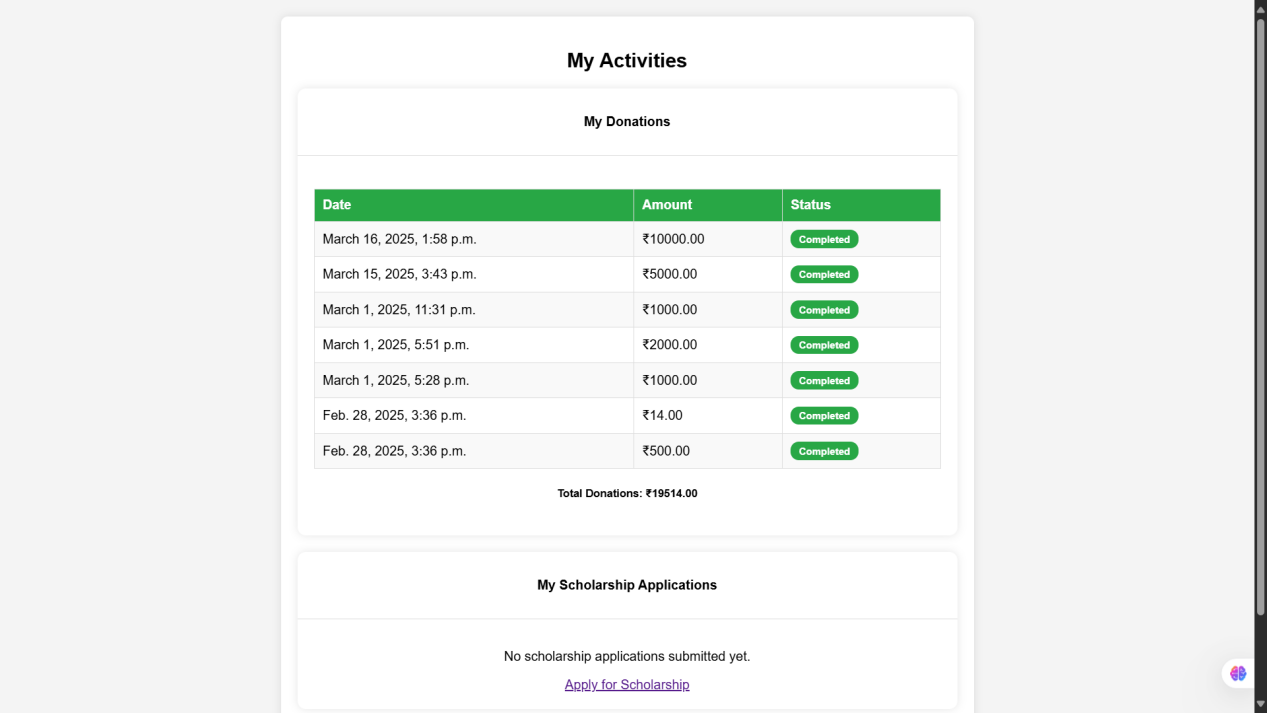
****

**Volunteer approval page**

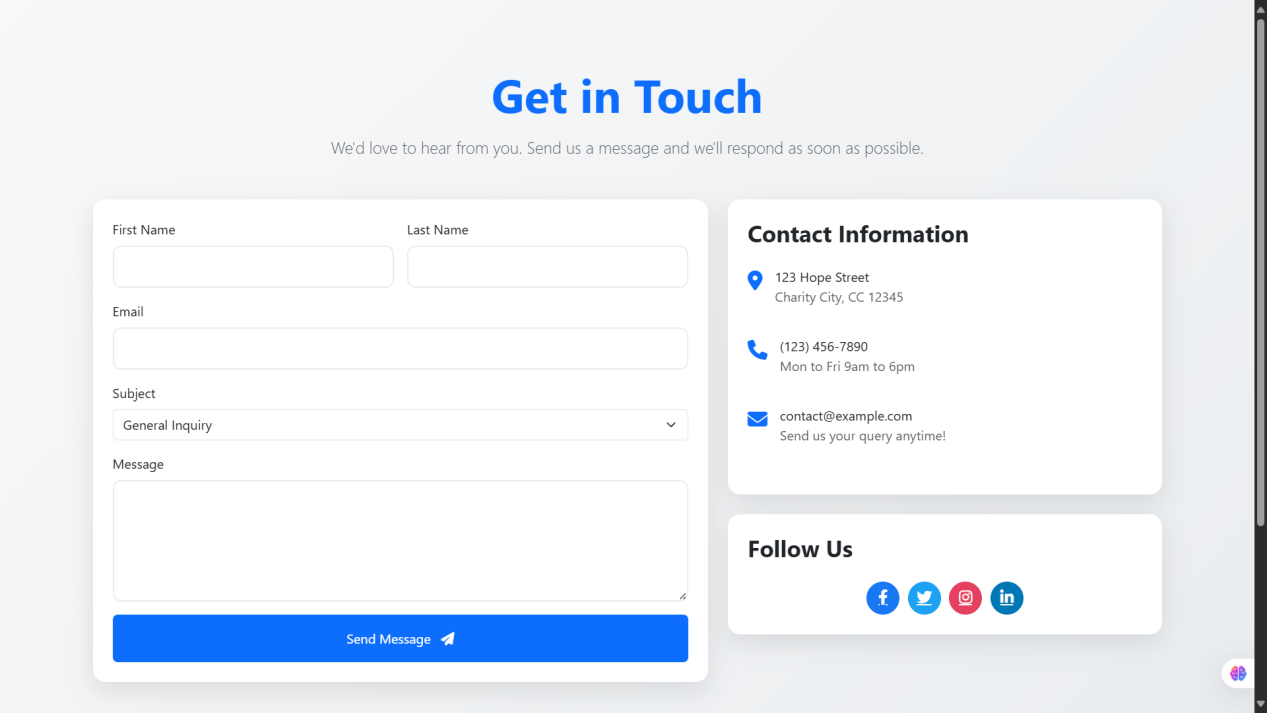
****

****

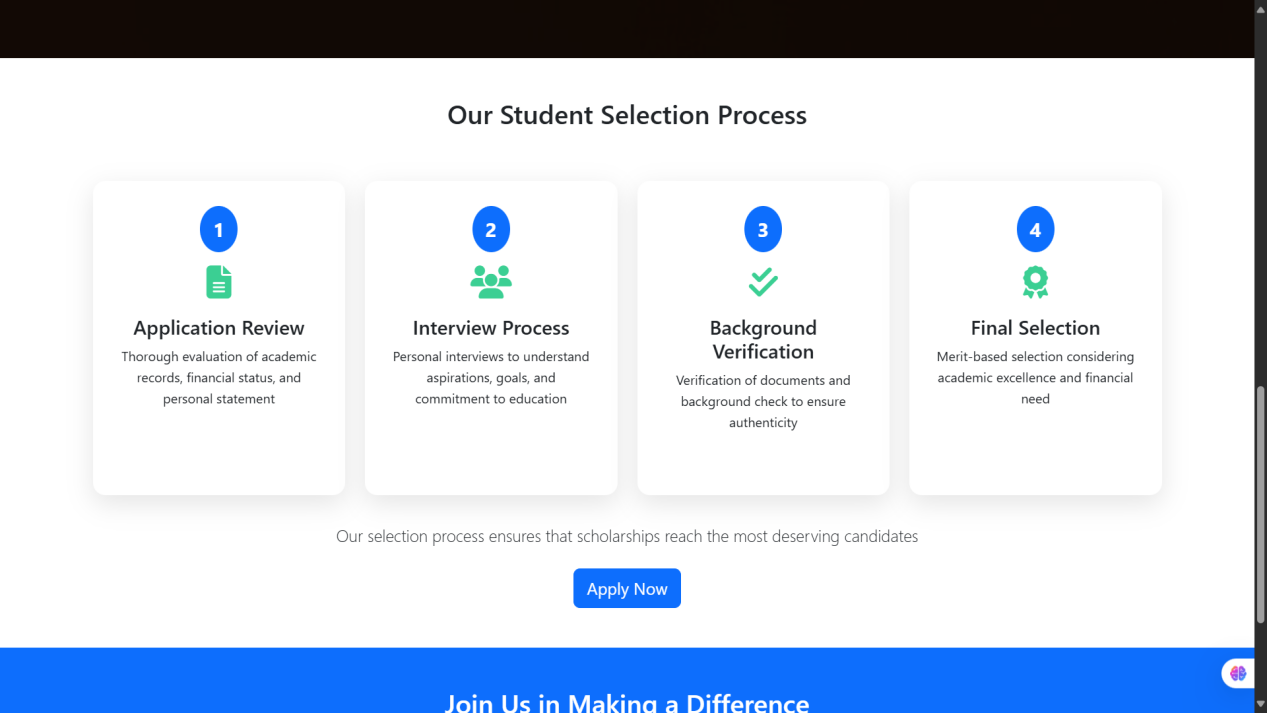
**My donation details**

****

**Message page**

****

**Student selection process**

****