**TEAM 21: A REPORT ON THE BURSARY BRIDGE PLATFORM DEVELOPMENT PROJECT**

Table of Content

[Introduction 2](#_Toc175617014)

[Project Overview: 2](#_Toc175617015)

[Purpose of the Report: 2](#_Toc175617016)

[Development Process 2](#_Toc175617017)

[1. Setting Up the Laravel 2](#_Toc175617018)

[2. Authentication and Authorization 3](#_Toc175617019)

[3. Bursary Application Feature 4](#_Toc175617020)

[4. Donor Incorporation 5](#_Toc175617021)

[5. MPESA Integration 6](#_Toc175617022)

[User Interface Screenshots 6](#_Toc175617023)

[Conclusion 8](#_Toc175617024)

[Project Summary 8](#_Toc175617025)

[Future Enhancements 8](#_Toc175617026)

**Table of figures**

[Figure 1:Database configuration 2](#_Toc175617041)

[Figure 2:Authentication controllers 3](#_Toc175617042)

[Figure 3:Authentication views 3](#_Toc175617043)

[Figure 4:Bursary Application code snippet 4](#_Toc175617044)

[Figure 5:Bursary Application Controller snippet 5](#_Toc175617045)

[Figure 6:Homepage snippet 7](#_Toc175617046)

[Figure 7:Login UI 7](#_Toc175617047)

[Figure 8:Register UI 8](#_Toc175617048)

[Figure 9:Bursary Application page 8](#_Toc175617049)

# Introduction

## Project Overview:

The Bursary Bridge platform aims to connect donors and students in need of financial aid, facilitating the provision of bursaries. The platform allows students to register, apply for bursaries, and manage their profiles. It also includes functionalities for managing donor participation and integrating real-time payment methods like MPESA.

## Purpose of the Report:

This report documents the development process of the Bursary Bridge platform, detailing the key features, challenges faced, and the solutions implemented. The report will include code snippets, screenshots, and technical explanations.

# Development Process

## 1. Setting Up the Laravel

**Installation and Configuration:**

The project was initiated by setting up a Laravel environment after installing it. Laravel's powerful MVC architecture facilitated structured development. The installation involved configuring environment variables to match the local development setup.

**Database Configuration:**

The MySQL database was configured with the necessary tables for users, bursary applications, and other relevant entities. This was done in the .env file and the database was integrated using Laravel's Eloquent ORM, which simplifies database interactions.

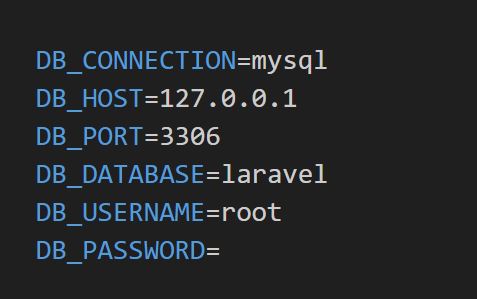


Figure :Database configuration

## 2. Authentication and Authorization

**Implementation:**

The project uses Laravel's built-in authentication starter kits. The `AuthenticatedSessionController` was customized to manage user logins, and additional features like email verification and password resets were implemented to enhance security.

**Challenges faced here:**

Styling Issues: Initially faced issues with customizing the login and register pages which was sorted out by fine-tuning the present code with some tailwind css.

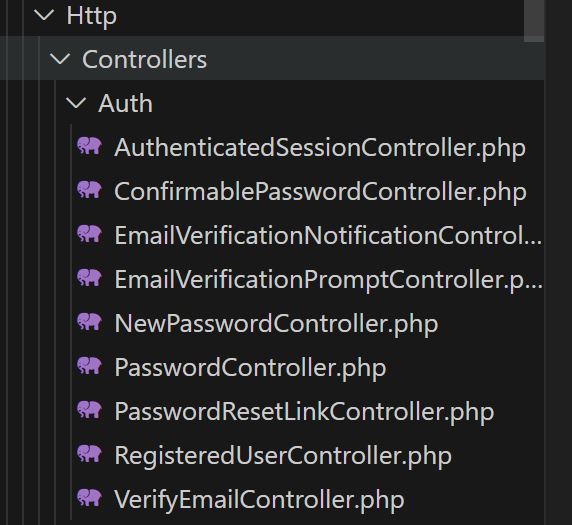


Figure :Authentication controllers

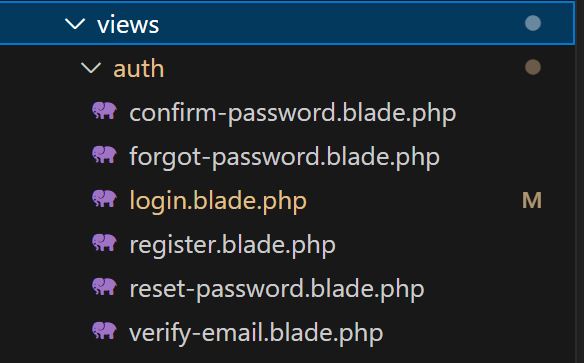


Figure :Authentication views

## 3. Bursary Application Feature

**Form Creation:**

The Bursary Application page was created with a form containing 20 fields. Only logged-in students can access this page. The form allows students to submit detailed information, including personal details, educational background, and financial need.

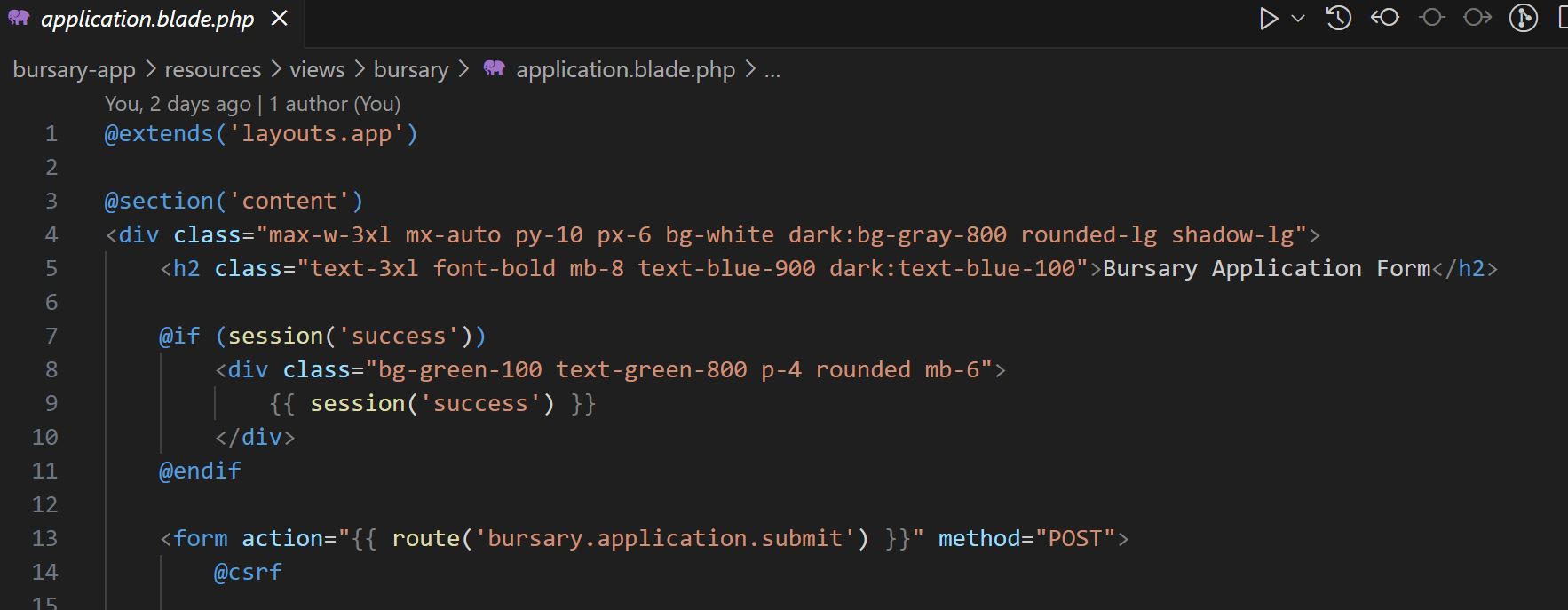


Figure :Bursary Application code snippet

**Controller and Validation:**

The Bursary Application Controller handles form submissions, implementing validation rules to ensure data integrity. Form validation was crucial in ensuring that only complete and correct information was stored in the database.

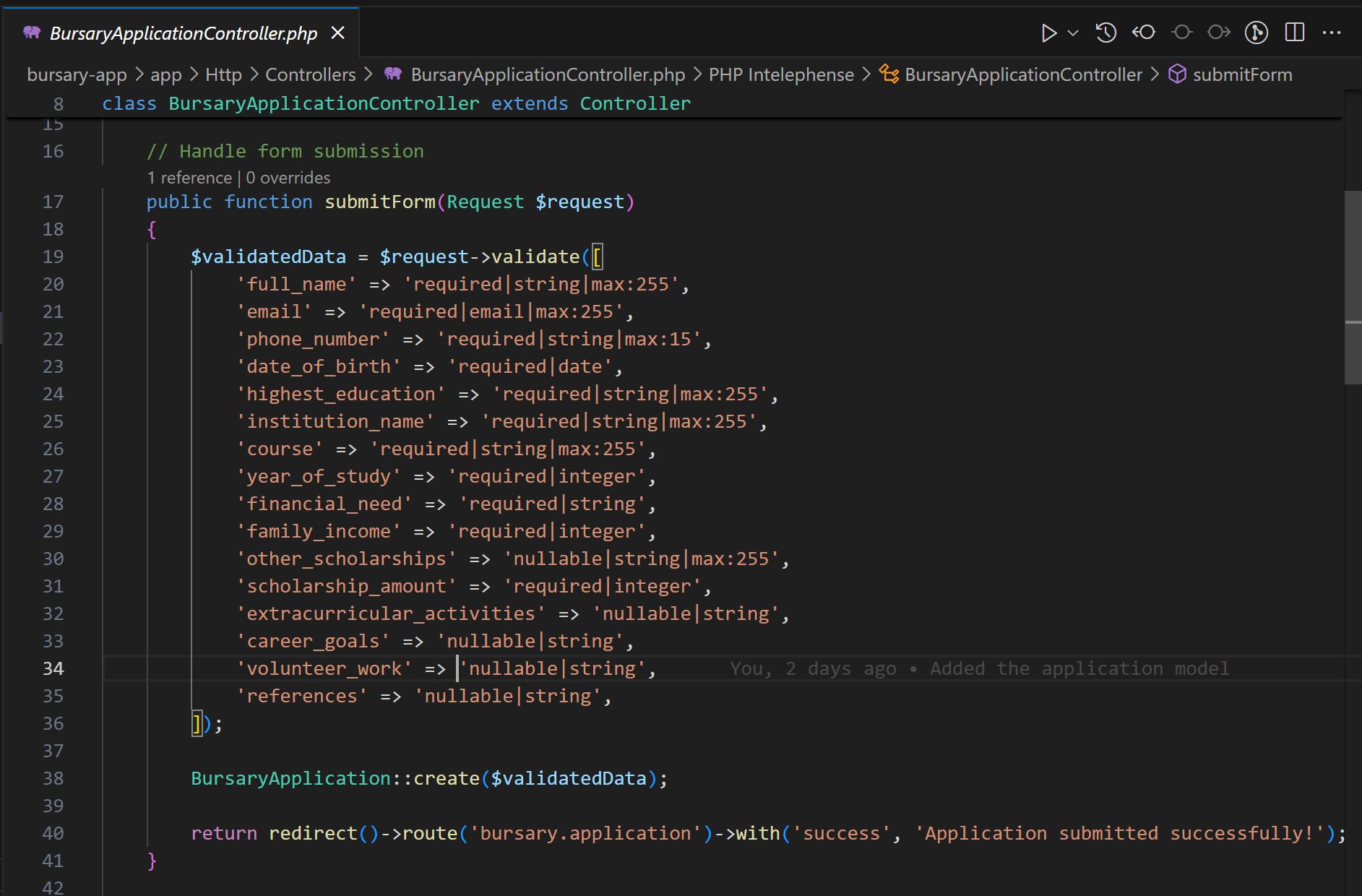


Figure :Bursary Application Controller snippet

**Styling:**

The Bursary Application page was styled with a blue theme consistent with the welcome page for a cohesive user experience.

**Challenges faced here:**

Validation Logic: Faced issues with validation requirements, which were resolved by breaking down validation rules and using custom error messages for better user feedback.

## 4. Donor Incorporation

**Design Considerations:**

The platform needed a mechanism for donors to either log in and directly provide bursaries or for administrators to manually input bursary details and receive donations. This design decision was crucial for ensuring flexibility in donor interactions.

**Implementation:**

After considering various approaches, it was decided that an administrator would manually input bursary details, simplifying the process while maintaining control over the data.

**Challenges faced here:**

Design Complexity: Figuring out if we need to give donors a page in which they view and track their interests or having an administrator in the middle managing that and designing the actual provisions for them in the app.

**Small code snippets for this include introducing a button in the welcome page and also creating a route for it as shown in the code below:**

//resources/views/welcome.blade.php

<a href="{{ url('/donate') }}" class="bg-blue-600 text-white px-6 py-3 rounded-md text-lg hover:bg-blue-700">

    Donate

</a>

// routes/web.php

Route::get('/donate', function () {

    return view('donate');

});

## 5. MPESA Integration

**API Integration:**

It involved: Registered the application with Safaricom to obtain the necessary API credentials, including the Consumer Key and Consumer Secret then configuring the API endpoint URLs in the Laravel .env file, ensuring secure handling of sensitive credentials.

Implemented the **Lipa na MPESA Online (STK Push)** functionality to facilitate real-time donations directly from users' mobile devices.

Developed a controller to handle payment requests, generate the necessary payment token, and initiate the STK Push request to Safaricom's API.

**Challenges:**

API Integration Issues:

Initial challenges in integrating the API were overcome by thoroughly reading the API documentation and correctly configuring API keys and endpoints.

# User Interface screenshots

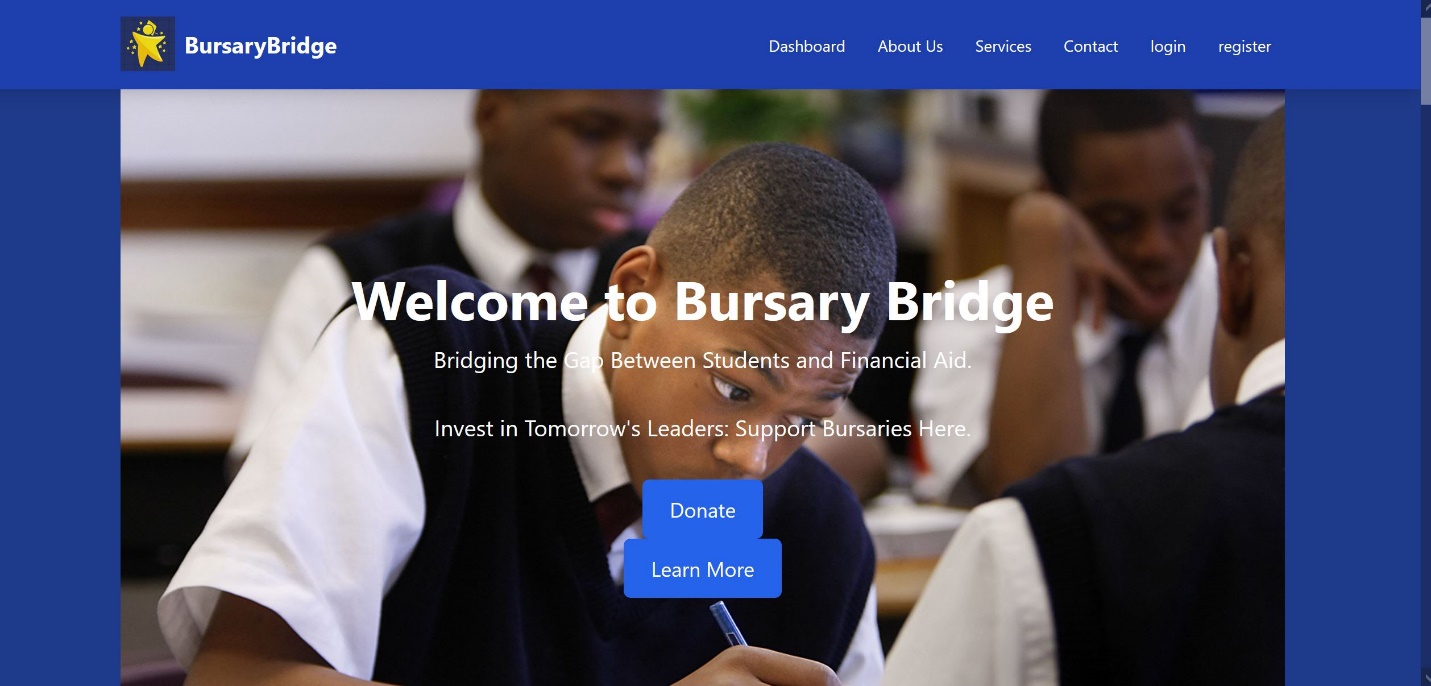


Figure :Homepage snippet

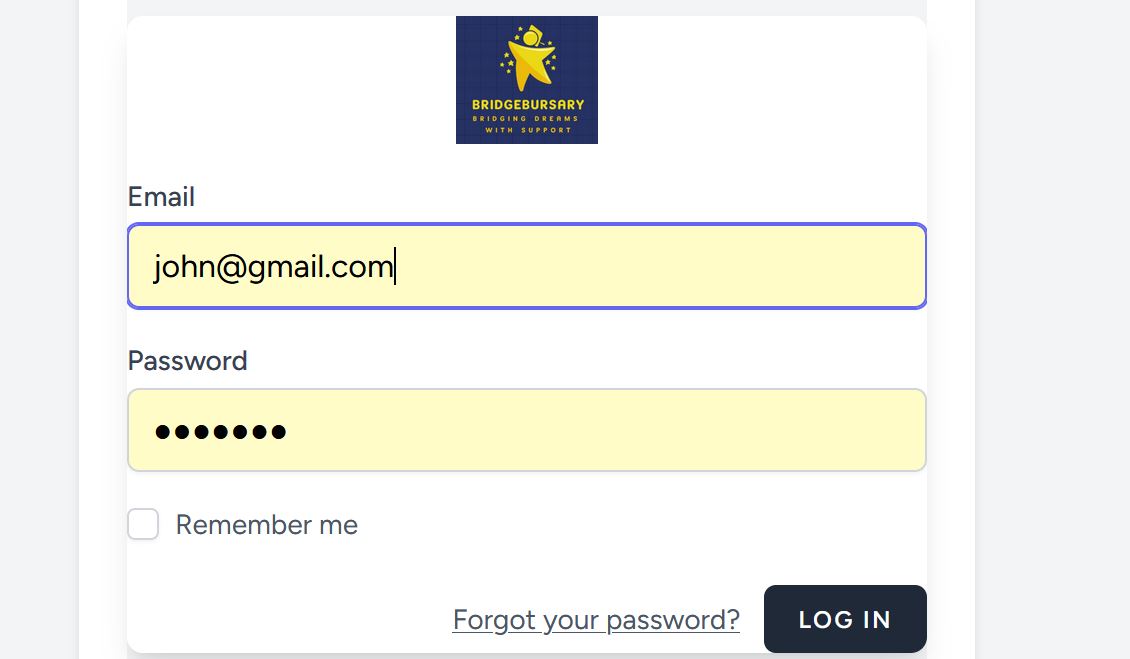


Figure :Login UI

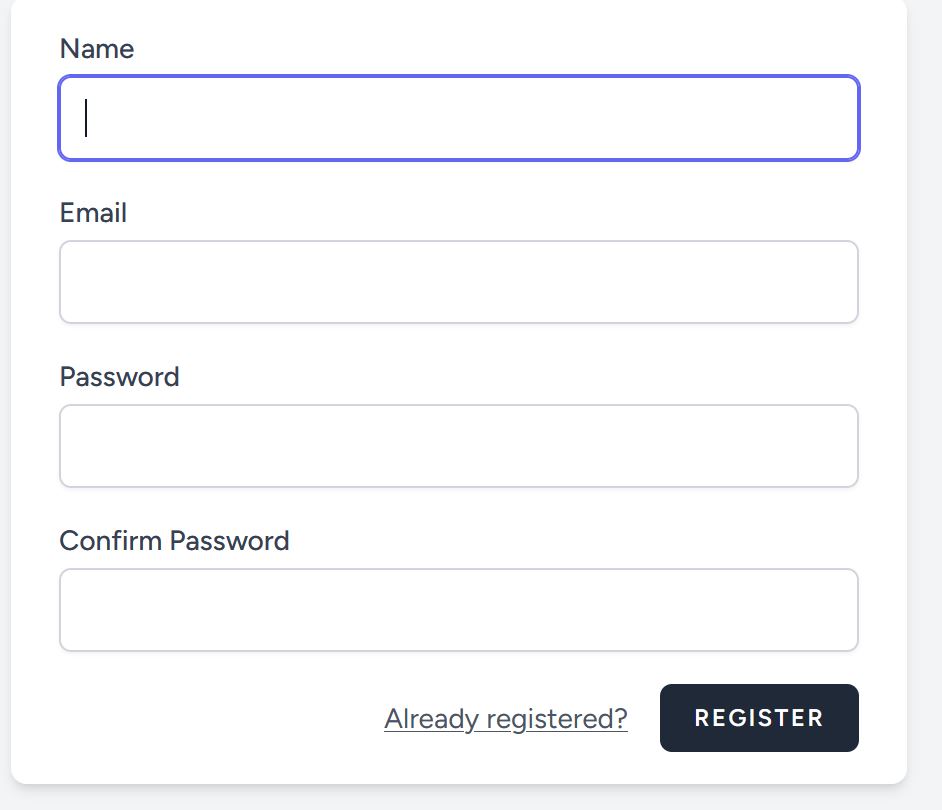


Figure :Register UI

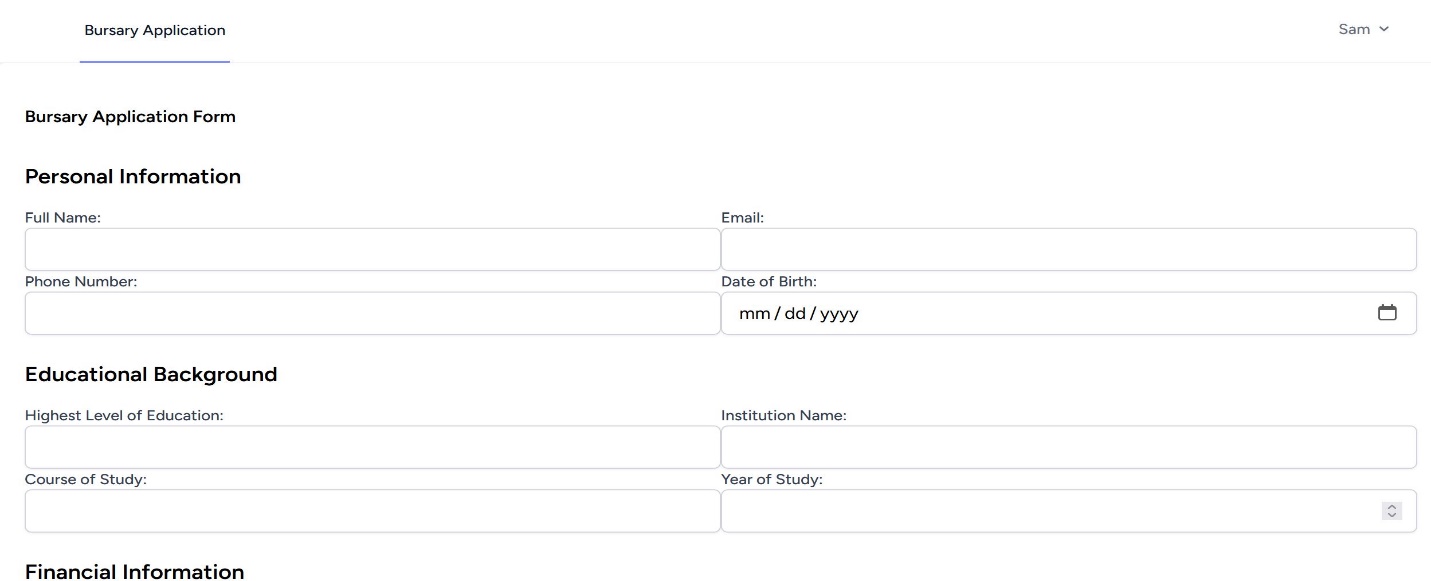


Figure :Bursary Application page

# Conclusion

## Project Summary

The project successfully implemented a comprehensive platform that allows students to register, apply for bursaries, and manage their profiles efficiently. Additionally, the system provides donors with a streamlined interface to view and contribute to bursaries, ensuring that financial aid is distributed effectively. The use of Laravel for backend development and Blade templating for the frontend contributed to the ease of development of a responsive application.

## Future Enhancements

**1. Advanced Donor Management System:**

- Implement a more sophisticated donor management system that includes detailed donor profiles, donation history, and personalized recommendations for bursaries based on the donor's preferences.

- Introduce a donor login feature that allows donors to manage their contributions, track the impact of their donations, and communicate with students they have supported.

**2. Real-Time Features:**

- Expand the platform to include more real-time features, such as live updates on bursary application statuses, instant notifications for both students and donors, and real-time tracking of donation goals.

- Integrate with additional payment gateways to facilitate instant donations and ensure seamless financial transactions.

**3. Expanded Analytics and Reporting:**

- Develop detailed analytics and reporting tools for both administrators and donors, providing insights into application trends, financial distributions, and the overall impact of the platform.

**4. Enhanced Security Measures:**

- Implement advanced security features, such as two-factor authentication for both students and donors, to ensure the highest level of data protection.

- Regularly audit the platform for security vulnerabilities and keep it updated with the latest security.