



TUNIS BUSINESS SCHOOL
UNIVERSITY OF TUNIS

OpportunityHub Web API Project Report

Mohamed Amine Hamadi

BA/IT student

IT-325 Web Services

Sunday 19th January, 2025

Abstract

Opportunity Hub is an API designed to enhance students' accessibility to scholarships and streamline their search process, with a specific focus on Tunisia. This API bridges the gap between Tunisian students, universities, and international program providers. It empowers program providers to share their scholarships and opportunities, while Tunisian universities can easily retrieve and disseminate these programs to their students. Beyond providing access to scholarships, Opportunity Hub is centered on improving and facilitating the student experience. Tunisian students can directly access opportunities that align with their eligibility criteria, eliminating the need to navigate international websites—a process often viewed as cumbersome and time-consuming. By offering a localized and user-friendly platform, Opportunity Hub simplifies and optimizes the process for students seeking educational and professional opportunities. Key features include tips on applying, reviews from alumni, country-specific documentation requirements, and scholarship procedures. The platform also fosters student networking by creating dedicated Discord discussions for each scholarship, where students can engage with alumni, ask questions, and collaborate.

Key Features

1. Centralized Platform for Programs:

- Provides a centralized platform where Tunisian students can find programs they are eligible for, with the ability to search using various filters.

2. Alumni Feedback:

- Students can get insights and advice from alumni and current scholarship recipients.

3. Real-Time Q&A and Networking:

- Offers discussion channels for students to ask questions, share experiences, and network.

4. Increased Accessibility:

- Ensures all Tunisian students have access to the same scholarship opportunities.

5. Student-Centered Community:

- Fosters a supportive, collaborative space for students to share experiences and advice.

Contents

1	Context and Background	1
1.1	Introduction	1
1.2	Context and Problem Statement	2
1.3	Scope and Objectives	2
1.4	Existing Work and Research	3
1.5	Originality and Contribution to the Field	3
2	Design and Contribution	4
2.1	Conceptual Design	4
2.1.1	Student	4
2.1.2	Program Providers	5
2.1.3	Tunisian Universities	5
2.1.4	UML Use Case Diagram	5
2.2	Implementation and Contribution	7
2.2.1	OAuth2 Contribution	7
2.2.2	HTTP Methods Contribution	8
2.2.3	PostgreSQL Contribution	8
2.2.4	Insomnia and Swagger Contribution	8
2.3	Database Design	10
2.3.1	Database Design	10
2.4	API Endpoints	12
2.4.1	Users Endpoints	12
2.4.2	Programs Endpoints	12
2.4.3	Reviews Endpoints	13
2.4.4	Discussions Endpoints	13
2.4.5	User Engagement Endpoint	14
2.4.6	Tips Endpoints	14
2.4.7	Requirements Endpoints	14
2.5	Discord API Integration	16
2.5.1	Purpose and Functionality	16
2.5.2	Technical Integration Overview	16
2.5.3	Benefits for Students	17
2.6	Security Measures Implementation	18
2.6.1	Password Hashing	18
2.6.2	Token-Based Authentication	18
2.6.3	Role-based access control	18
2.6.4	Token Refresh and Revocation	18
2.6.5	Password Reset and Update	18

3	Conclusion and Potential Improvements	19
3.1	Conclusion	19
3.2	Potential Improvements	19
3.3	References	20

List of Figures

2.1	Use Case Diagram	6
2.2	Postgres	8
2.3	Insomnia for Testing OpportunityHub API	8
2.4	Swagger documentation for OpportunityHub API	9
2.5	Database Design	11
2.6	Discord channel for Thomas Jefferson Scholarship	17

Chapter 1

Context and Background

1.1 Introduction

In Tunisia, the interest in scholarships has been growing significantly among students, particularly in recent years. As more students seek opportunities to study abroad or pursue advanced degrees, scholarships have become an essential part of their academic journey. However, many of these students face significant challenges during the scholarship application process, which can be discouraging and overwhelming. One of the primary issues is that students often do not know where to look for relevant scholarship opportunities. **Tunisian universities** typically do not provide comprehensive lists of scholarships on their websites, which leaves students searching through a multitude of international websites—most of which do not specify whether Tunisians are eligible to apply. The sheer number of these global platforms, combined with the lack of clear eligibility criteria, makes the search process complex and time-consuming.

Additionally, even after students find scholarships, they often face further challenges related to the application process itself. Common concerns include understanding the specific requirements of each program, the steps involved in the application, country-specific documentation procedures, and the overall experience during the scholarship period. Unfortunately, these questions frequently go unanswered, or are only addressed in a vague and insufficient manner, leaving students feeling uncertain and anxious about the process.

This is where the **Opportunity Hub API** steps in to solve these problems. Opportunity Hub aims to create a unified platform that connects Tunisian students, program providers, and tunisian universities, thereby streamlining the entire process. Through this API, students will gain access to a comprehensive database of scholarships specifically available to Tunisians, published by partnering program providers. **Tunisian universities**, which currently do not provide sufficient scholarship information, can retrieve and share the available opportunities through Opportunity Hub, making it easier for students to access the programs they are eligible for without the need to navigate countless websites.

In addition to simplifying the search for scholarships, Opportunity Hub will offer a much-needed feature for students: realistic feedback from **Alumni** who have previously participated in these scholarship programs. By allowing students to contact those who have already gone through the experience, Opportunity Hub provides valuable insights into what the program is like, how to apply, and what to expect. This peer-to-peer exchange will help students make more informed decisions and alleviate some of the

uncertainties that come with applying for scholarships.

Furthermore, Opportunity Hub will foster a sense of community among Tunisian students by creating a space for them to network and discuss scholarship opportunities. The platform will facilitate discussion groups for specific scholarships through **Discord channels**, where students can ask questions, share their experiences, and connect with others who are on a similar journey. This collaborative environment will not only provide practical support but also offer emotional encouragement, making students feel more confident and empowered throughout the scholarship application process.

In conclusion, Opportunity Hub is designed to address the various challenges faced by Tunisian students in accessing and applying for scholarships. By providing a centralized platform for scholarship information, facilitating peer-to-peer support, and offering networking opportunities, the API will empower students to take full advantage of the scholarships available to them, ultimately contributing to their academic success and personal growth.

1.2 Context and Problem Statement

In Tunisia, many students are interested in exchange programs. However, they encounter significant challenges in identifying relevant opportunities. Tunisian universities **lack centralized platforms** that aggregate available scholarships, forcing students to navigate numerous international websites. Many of these sites do not specify whether Tunisians are eligible to apply, making the search process overwhelming and discouraging. Additionally, students often have unresolved questions regarding the **application process, country-specific documentation requirements, and the program experience itself**. These uncertainties, coupled with the lack of clear guidance, prevent many students from fully exploring and seizing opportunities. This gap highlights the need for a **comprehensive solution to assist students in navigating scholarship options and improving their academic prospects**.

1.3 Scope and Objectives

The **Opportunity Hub API** is designed to bridge the gap between Tunisian students and program providers. The scope of the project is to create a centralized platform where Tunisian students can easily access programs. The objectives of this project are to:

- Provide a comprehensive database of scholarships that are specifically available to **Tunisian students**.
- Enable students to engage with alumni and current scholarship recipients to get real-world insights, tips on how to apply, and reviews.
- Foster a collaborative space for students to discuss specific scholarships and share experiences.
- Provide detailed information on the requirements of each scholarship, along with country-specific documentation requirements, so students can be well-prepared for visa applications and other program-specific procedures.

By addressing these objectives, the platform aims to reduce the barriers faced by students and make the scholarship application process more accessible and efficient.

1.4 Existing Work and Research

Several existing platforms and websites provide program information, but many of them are international and lack specific focus on the Tunisian context. Platforms such as **ScholarshipPortal**, **Chevening**, and **DAAD** offer global opportunities, but they do not tailor their content to Tunisian students, and eligibility for each scholarship is not always clear. Additionally, these websites do not provide localized support or peer-to-peer networks, which can significantly help students navigate the application process.

In Tunisia, the Ministry of Higher Education occasionally posts scholarship opportunities on its website. However, this section is not a primary focus of the site and is rarely updated. The information is posted sporadically and lacks clear explanations or detailed descriptions of the opportunities. Furthermore, many Tunisian students are unaware of this section, as it is not designed to attract their attention. Without engaging features or comprehensive support, the Ministry's website fails to serve as an effective resource for students looking for scholarships.

Thus, there is a significant gap in the Tunisian market for a centralized platform that aggregates relevant scholarships, provides localized support, and includes features like feedback, networking, and real-time discussions. OpportunityHub aims to fill this gap by offering a comprehensive, user-friendly platform tailored specifically to the needs of Tunisian students.

1.5 Originality and Contribution to the Field

Opportunity Hub is a pioneering platform tailored to the specific needs of Tunisian students, offering an innovative solution in the realm of scholarship access and exchange programs. Unlike global platforms that cater to a wide range of international students, Opportunity Hub is designed exclusively to address the unique challenges faced by students in Tunisia. The platform's originality lies in its localized approach, focusing only on opportunities relevant to Tunisian students and integrating features that directly support them in their scholarship journey.

The platform's key innovation is its emphasis on community-driven engagement, allowing students to ask questions, seek advice from alumni, and receive guidance on the application process. Opportunity Hub enables students to interact with past scholarship recipients, providing them with realistic reviews, insights, and tips on how to navigate their applications. By incorporating real-time Q&A features, such as Discord channels, students can engage directly with alumni and peers, fostering a network of support that goes beyond traditional informational resources. This peer-to-peer connection is essential in helping students make informed decisions about which scholarships to apply for and how to best present themselves in their applications.

Opportunity Hub also stands out by providing detailed country-specific documentation requirements, an often-overlooked yet essential aspect of applying for international scholarships. By offering clear, accessible information on these requirements, the platform helps students avoid confusion and ensures they are well-prepared for visa applications and other formalities that can otherwise be a barrier to success.

Through its innovative features and localized focus, Opportunity Hub significantly improves the scholarship ecosystem in Tunisia, fostering better communication and collaboration.

Chapter 2

Design and Contribution

2.1 Conceptual Design

The **Opportunity Hub API** connects three key actors: **Students**, **Program Providers**, and **Tunisian Universities**. Each actor has unique responsibilities and accesses different features of the platform, facilitating the connection between Tunisian students and opportunities.

2.1.1 Student

Students are the primary users of the platform. They interact with the Opportunity Hub to explore and apply for scholarships. Their responsibilities include:

- **Create an Account:** Students can register and create a profile to access personalized scholarship information.
- **Browse Available Scholarships:** Students can explore scholarships that eligible to Tunisian students, based on criteria such as funding type, field of study, and location.
- **Join or Create Discussions:** Students can join or create discussion groups related to specific scholarships on Discord, fostering community engagement and collaboration.
- **Mark Scholarships of Interest:** Students can mark scholarships they are interested in or have applied to, enabling them to track their progress.
- **Alumni Reviews:** After participating in a scholarship program, students can leave reviews, providing valuable insights for other students considering the same opportunities.
- **Engage with Alumni:** Students can access feedback and contact details from previous scholarship recipients to directly ask questions about their experiences.
- **Add Tips on How to Apply:** Students can add tips and advice on how to apply for specific scholarships, helping future applicants with guidance and personal insights.

2.1.2 Program Providers

Program providers are organizations or institutions offering scholarships through the Opportunity Hub API. Their responsibilities include:

- **Create an Account:** Providers can register an account to gain the ability to share scholarship opportunities.
- **Create and Manage Scholarship Listings:** Providers can create new scholarship entries, including details like eligibility criteria, application process, deadlines, and more.
- **Access All Listed Programs:** Providers can view all the programs that have been listed on the platform.

2.1.3 Tunisian Universities

Universities serve as facilitators, helping their students find relevant scholarships. Their roles include:

- **Create an Account:** Universities can register as partners to facilitate their access to available scholarships.
- **Retrieve and Share Scholarships:** Universities can access the list of scholarships and share them on their websites, ensuring that students are informed about opportunities that are relevant to them.
- **Monitor Student Interest:** Universities can track the number of students interested in specific scholarships and promote particular programs to encourage participation.
- **Add Country-Specific Document Requirements:** Universities can add and manage country-specific document requirements for scholarships, ensuring that students are aware of the necessary paperwork for applying to each opportunity.

2.1.4 UML Use Case Diagram

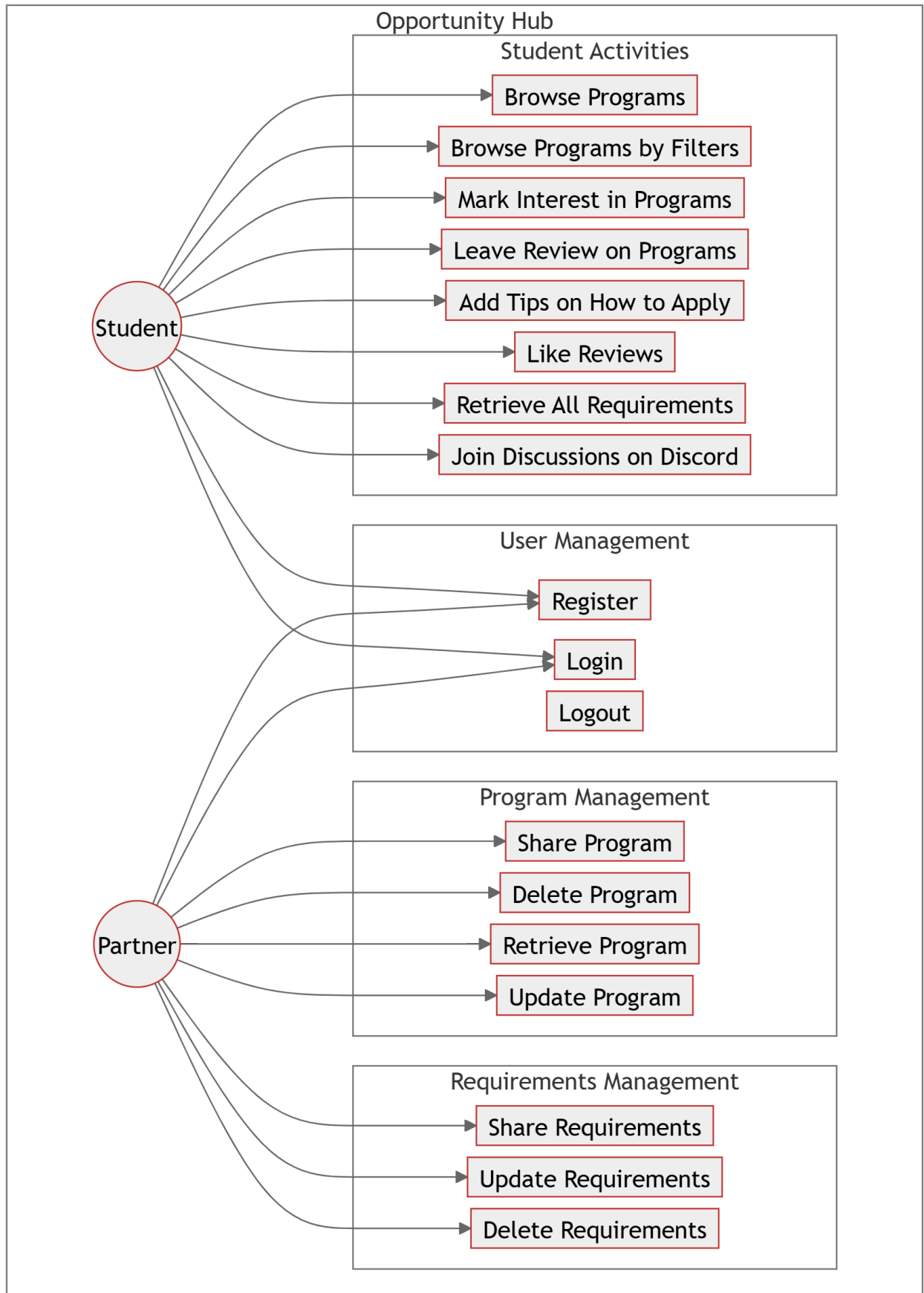


Figure 2.1: Use Case Diagram

2.2 Implementation and Contribution

We introduce the creation and deployment of a Web API using **FastAPI**, **Python**, **SQLAlchemy**, and **PostgreSQL**. The selection of these technologies was driven by various considerations.

To begin with, **FastAPI** stands out as a contemporary, swift, and effective framework designed for API development. Leveraging the **Python** standard library, it ensures ease of use for developers familiar with **Python**. Moreover, it is constructed upon **Starlette**, an asynchronous web framework, enabling high performance and scalability.

Python, being a widely adopted and versatile programming language, is well-suited for a broad spectrum of applications, including web development. Its extensive ecosystem of libraries and frameworks makes it an excellent choice for crafting intricate and feature-rich applications.

SQLAlchemy, a renowned Object-Relational Mapping (ORM) library for Python, simplifies database interactions, offering developers a more convenient and efficient approach to working with data. By abstracting the complexities of database interactions, it provides a straightforward, **Pythonic API**.

Lastly, **PostgreSQL**, an open-source relational database management system, was chosen for its robustness, scalability, and performance. Its reliability and feature-rich nature make it an apt selection for managing and storing data in the context of the Gaming API project.

Below is a list of most important packages and frameworks used for the project:

- fastapi
- psycopg2
- pydantic
- SQLAlchemy
- uvicorn
- alembic

2.2.1 OAuth2 Contribution

Implemented to facilitate secure authentication, enhancing user login options and overall account security.

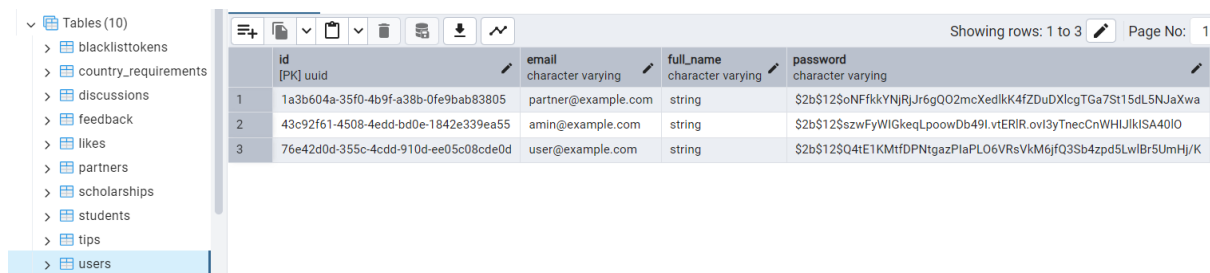
```
oauth2_scheme = OAuth2PasswordBearer(tokenUrl='login' )
```

2.2.2 HTTP Methods Contribution

Leveraged various HTTP methods (GET, POST, PUT, DELETE) to enable standard RESTful operations for managing gaming sessions, providing a comprehensive API.

2.2.3 PostgreSQL Contribution

Chosen as the relational database management system to store and manage gaming session data efficiently.



	id [PK] uuid	email character varying	full_name character varying	password character varying
1	1a3b604a-35f0-4b9f-a38b-0fe9bab83805	partner@example.com	string	\$2b\$12\$SoNffkYnJRjRégQO2mcXedlKk4fZDuDXlCgTGa7St15dL5NJaXwa
2	43c92f61-4508-4edd-bd0e-1842e339ea55	amin@example.com	string	\$2b\$12\$szwFyWIGkeqLpoowDb49I.vtERIR.ov3yTnecCnWHlJlkISA40IO
3	76e42d0d-355c-4cdd-910d-ee05c08cde0d	user@example.com	string	\$2b\$12\$Q4tE1KMtFDpNtgazPlaPLO6VRsVikM6jfQ3Sb4zpd5LwIBr5UmHj/K

Figure 2.2: Postgres

2.2.4 Insomnia and Swagger Contribution

Employed for API testing and documentation, streamlining the development process and ensuring clear communication of endpoints and functionalities.

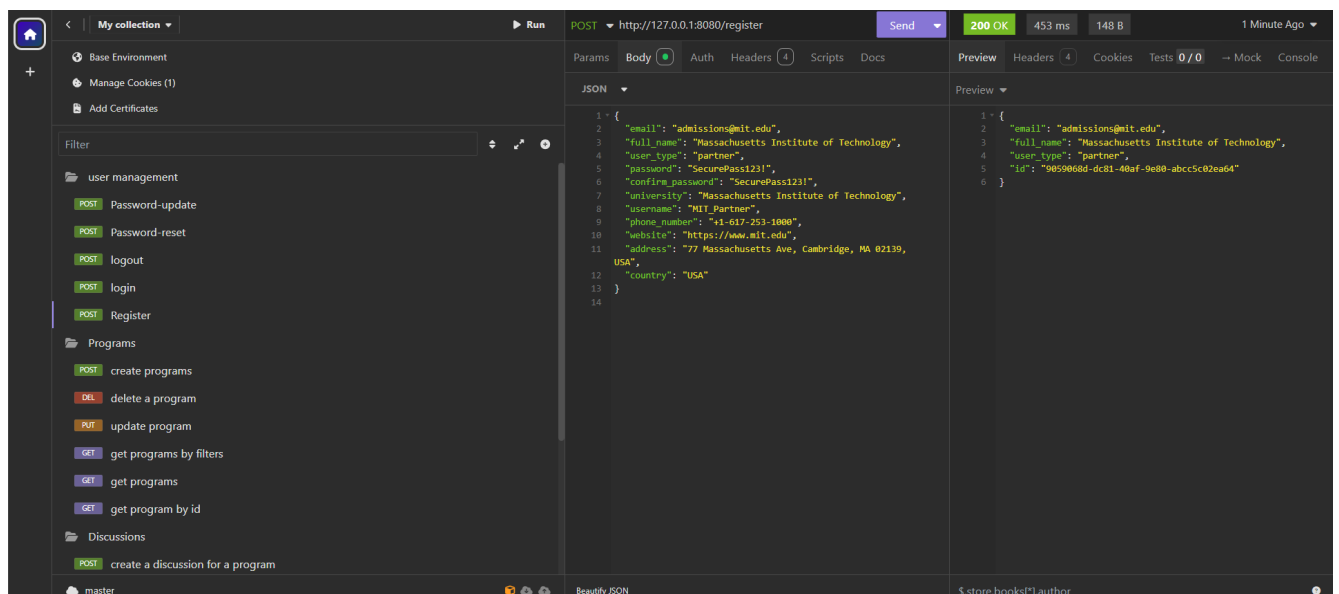


Figure 2.3: Insomnia for Testing OpportunityHub API

users			^
POST	/register	Register	✓
POST	/login	Login	✓
POST	/refresh	Refresh	✓
POST	/logout	Logout	🔒 ✓
POST	/password-reset	Password Reset Token	✓
POST	/password-update	Password Update	🔒 ✓
Programs			^
POST	/Programs/	Create Scholarship	🔒 ✓
GET	/api/Programs	Get Scholarships	✓
GET	/api/Programs/filters	Get Scholarships By Filters	✓
GET	/api/Programs/{id}	Get Scholarship	✓
DELETE	/api/Programs/{id}	Delete Scholarship	🔒 ✓
PUT	/api/Program/{id}	Update Scholarship	🔒 ✓
Reviews			^
POST	/Reviews/	Create Feedback	🔒 ✓
GET	/api/Reviews/{scholarship_id}	Get Feedback	✓
DELETE	/api/Reviews/{id}	Delete Feedback	🔒 ✓
POST	/Reviews/{feedback_id}/like	Add Like	🔒 ✓
POST	/Programs/mark-interest/{scholarship_id}	Add Interest	🔒 ✓
Discussions			^
POST	/create-channel/	Create Channel	🔒 ✓
GET	/discussions/{scholarship_id}	Get Discussion By Scholarship Id	🔒 ✓
Requirements			^
POST	/requirements	Add Country Requirement	🔒 ✓
GET	/requirements/{country}	Get Requirements By Country	✓
PUT	/requirements/{requirement_id}	Update Country Requirement	🔒 ✓
DELETE	/requirements/{requirement_id}	Delete Country Requirement	🔒 ✓
Tips			^
POST	/tips	Create Tip	🔒 ✓
GET	/tips/{scholarship_id}	Get Tips By Scholarship	✓
PUT	/tips/{tip_id}	Update Tip	🔒 ✓

Figure 2.4: Swagger documentation for OpportunityHub API

2.3 Database Design

2.3.1 Database Design

The database is stored using PostgreSQL and has the following tables:

- **Table Partners:**
 - Stores the information of the Partners.
- **Table Students:**
 - Stores the information of the Students.
- **Table Programs:**
 - Stores information of the available Programs. It has `partner_id` as a foreign key from table Partners.
- **Table Reviews:**
 - Stores the reviews written by Students. It has `student_id` as a foreign key from table Students and `program_id` as a foreign key from table Programs.
- **Table Likes:**
 - Stores the likes given by Students on Reviews. It has `student_id` as a foreign key from table Students and `review_id` as a foreign key from table Reviews.
- **Table Discussions:**
 - Stores discussions related to Programs. It has `program_id` as a foreign key from table Programs and `student_id` as a foreign key from table Students.
- **Table Country_Requirements:**
 - Stores the required documents for each country, including document type, description, and whether the document is mandatory or not.
- **Table Tips:**
 - Stores tips provided by Alumni for specific scholarships, with `student_id` and `program_id` as foreign keys.

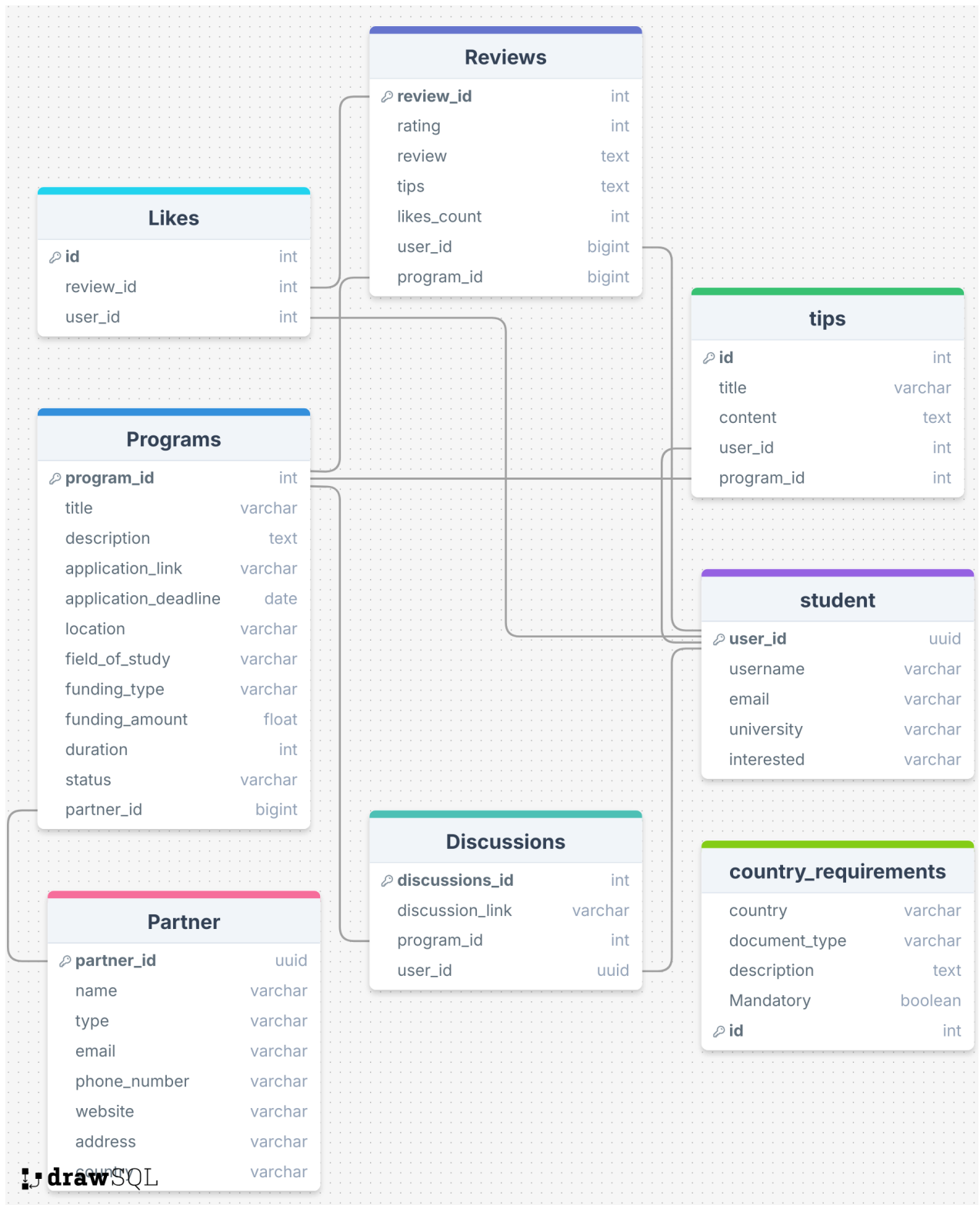


Figure 2.5: Database Design

2.4 API Endpoints

2.4.1 Users Endpoints

- **/register (POST):** This endpoint allows users to register by providing their email, password, and other optional details. If the email is already registered, it returns a 400 error. After hashing the password, the user is created as either a student or partner, based on the `user_type`. A student record is created with additional details such as university, while a partner record is created with contact information like phone number, website, and address. Upon successful registration, the endpoint returns the newly created user's information.
- **/login (POST):** This endpoint allows users to log in by providing their username or email and password. If the provided credentials are incorrect, it returns a 400 error. Upon successful authentication, a token pair (access token and refresh token) is generated. The refresh token is stored as a secure cookie, and the access token is returned in the response with a "bearer" token type.
- **/refresh (POST):** This endpoint allows users to refresh their access token by providing a valid refresh token in the cookies. If no refresh token is found, it returns a 400 error with the message "refresh token required". Upon receiving a valid token, the endpoint generates and returns a new access token.
- **/logout (POST):** This endpoint allows users to log out by providing a valid access token. The token is decoded to retrieve the payload, and the token is added to a blacklist to invalidate it. Upon successful logout, a message "Successfully logged out" is returned.
- **/password-reset (POST):** This endpoint allows users to reset their password by providing a valid reset token and the new password. The token is decoded to verify the user's identity. If the user is found, the new password is hashed and updated in the database. A message indicating the successful update of the password is returned.
- **/password-update (POST):** This endpoint allows users to update their password by providing the old password, new password, and a valid access token. The old password is verified against the stored password, and if incorrect, a validation error is raised. If the old password is valid, the new password is hashed and updated in the database. A message confirming the successful password update is returned.

2.4.2 Programs Endpoints

- **/Programs/ (POST):** This endpoint allows a partner to create a new scholarship. The user must be authenticated and have a "partner" role. It retrieves the partner associated with the current user, and if the partner is found, a new scholarship is created and associated with the partner. The scholarship details (title, description, location, funding, etc.) are stored in the database, and the created scholarship is returned as a response.
- **/api/Programs (GET):** This endpoint retrieves a list of all available scholarships from the database. It executes a query to fetch all scholarships and returns them in

the response. If an error occurs during the query execution, a 500 Internal Server Error is raised with the relevant error message.

- **/api/Programs/id (GET):** This endpoint retrieves a specific scholarship program based on the provided ID. It performs an asynchronous query to the database and returns the scholarship details. If no scholarship is found with the given ID, it returns a 404 Not Found error along with an appropriate error message.
- **/api/Programs/id (DELETE):** This endpoint allows the deletion of a scholarship by its ID. It first checks whether the scholarship exists; if not, it returns a 404 Not Found error. If the deletion is successful, it responds with a 204 No Content status and no response body. In case of any unexpected errors, it raises a 500 Internal Server Error.
- **/api/Programs/filters (GET):** This endpoint allows users to retrieve scholarships based on various filters such as location, field_of_study, and funding_type. It checks for any provided filter parameters and applies them to the query. If no filters are provided, all scholarships are returned. The results are returned as a list of scholarships matching the provided criteria. If no matching scholarships are found, it will return an empty list.
- **/api/Programs/id (PUT):** This endpoint allows users to update an existing scholarship with the specified id. The request requires the scholarship_data object to provide the updated values. It ensures that the user is a partner and that the scholarship belongs to the partner attempting the update. If the scholarship is not found or the user is not authorized to update it, an appropriate error response will be returned. If successful, the scholarship details are updated, committed, and returned in the response.

2.4.3 Reviews Endpoints

- **/review/ (POST):** This endpoint allows students to submit feedback on a scholarship. The request requires the feedback_data object, which includes the scholarship ID, review... The system verifies that the current user is a student and that the provided scholarship exists. If both checks pass, the feedback is saved and associated with the student and scholarship.
- **/reviews/scholarship_id(GET):** This endpoint allows fetching all reviews for a specific scholarship (program) using its ID. It checks whether the scholarship exists. If the scholarship is found, it retrieves and returns all reviews associated with it. If the scholarship doesn't exist, it returns a 404 error.
- **/review/review_id(DELETE):** This endpoint allows users to delete their own reviews. It checks if the review exists and whether the current user is the one who created the review. If either condition is not met, it raises an error.

2.4.4 Discussions Endpoints

- **/create-channel(POST):** This endpoint allows for the creation of a Discord discussion channel linked to a specific scholarship. If a channel for the given scholarship

already exists, it retrieves and returns the link to the existing channel. If not, it creates a new text channel on Discord using the provided scholarship title, and then stores the new channel's ID in the database. The user is also associated with the channel, and a link to the new Discord channel is returned. Errors are raised if the Discord API fails during the process.

- **/discussions/scholarship_id(GET)**: This endpoint retrieves the Discord discussion link for a specific scholarship. It first checks if the scholarship exists in the database. If the scholarship is found, it then looks for the associated discussion channel. If a discussion is found, it constructs the link to the relevant Discord channel and returns it along with the scholarship title. If no scholarship or discussion is found, appropriate error messages are returned.

2.4.5 User Engagement Endpoint

- **/Reviews/{feedback_id}/like (POST)**: This endpoint allows a student to like a specific feedback for a scholarship. The request requires the `feedback_id` as a parameter. The system checks if the feedback exists and if the student has already liked it. If not, the like is added to the database, and the `likes_count` for the feedback is incremented. Upon success, the system returns a confirmation message with the updated `likes_count`.
- **/Programs/mark-interest/{scholarship_id} (POST)**: This endpoint allows a student to mark interest in a specific scholarship program. The request requires the `scholarship_id` as a parameter. Upon successful marking of interest, the student's record is updated, and a confirmation message is returned. The endpoint ensures the student is authenticated and exists in the system.

2.4.6 Tips Endpoints

- **/tips (POST)**: This endpoint allows users to create and share tips related to specific scholarships. The request body requires a title, content, and scholarship ID. Only students are authorized to create tips. Upon successful creation, the tip is added to the database, and a confirmation message with the `tip_id` is returned.
- **/tips/{scholarship_id} (GET)**: This endpoint retrieves all tips associated with a specific scholarship, identified by its `scholarship_id`. If the scholarship exists, all relevant tips are returned. If no tips are found, a 404 error is returned with a message indicating that no tips are available for that scholarship.
- **/tips/{tip_id} (PUT)**: This endpoint allows users to update an existing tip. Only the user who created the tip can update it. The request requires the `tip_id` and any fields to be updated (title or content). If the tip is found and the user is authorized, it will be updated in the database.

2.4.7 Requirements Endpoints

- **/requirements (POST)**: This endpoint allows users to add country-specific document requirements for scholarships. The request requires the country, document

type, description, and whether the document is mandatory. The user must be authorized as a partner. Upon successful creation, the new requirement is stored, and a confirmation message with the country is returned.

- **/requirements/{country} (GET):** This endpoint retrieves the list of document requirements for a specific country. The country is provided as a parameter, and the system checks if any requirements exist for that country. If no requirements are found, a 404 error is returned with a relevant message.
- **/requirements/{requirement_id} (PUT):** This endpoint allows users to update an existing country-specific requirement. Only users with "partner" privileges can perform this update. The request requires the `requirement_id` and the fields to be updated (e.g., document type, description). If the requirement is found and the user is authorized, it will be updated in the database.
- **/requirements/{requirement_id} (DELETE):** This endpoint allows users to delete a specific country requirement by its `requirement_id`. The user must be authorized as a partner. Upon successful deletion, a confirmation response is returned with the deleted requirement details.

2.5 Discord API Integration

To enhance communication and collaboration among students applying for the same scholarship, a Discord channel integration has been implemented. This integration allows students to interact with each other, ask questions, receive tips from alumni on applying, and network during their application process. The integration leverages the Discord API to create dedicated channels within a Discord server specifically for each scholarship. This provides a seamless and interactive platform for students to engage, discuss scholarship details, share experiences, and collaborate.

2.5.1 Purpose and Functionality

The main goal of the Discord API integration is to create a dedicated online space for students to interact. Upon applying for a scholarship, students can access a unique discussion channel where they can:

- **Ask Questions:** Students can ask questions about the scholarship, its requirements, deadlines, and other related topics.
- **Share Experiences:** Alumni can share their personal experiences, tips, and resources that may help other students.
- **Network:** The channel provides an opportunity for students to connect, build professional relationships, and potentially collaborate on common goals.

By using Discord, which is a popular communication platform, students have access to a familiar, user-friendly interface that promotes active engagement. With features like direct messaging, voice channels, and real-time chat, the platform ensures that students have the tools they need to connect with peers effectively.

2.5.2 Technical Integration Overview

The integration with Discord's API is managed through a FastAPI backend, which handles requests to create discussion channels based on specific Programs. The steps for this process are outlined below:

1. **Channel Creation:** When a new scholarship is created, a corresponding discussion channel is generated in a predefined Discord server using the Discord bot's credentials. This is done via the Discord API by sending a POST request that contains details about the scholarship.
2. **Channel Management:** If a channel already exists for a given scholarship, the system will simply return the existing channel's link, preventing duplicate channels. This ensures that each scholarship has a single, dedicated discussion space.
3. **Bot Authentication:** The bot is authenticated with Discord through a secure token, which is stored privately to avoid any unauthorized access.
4. **Database Integration:** Information about the created channels, including the channel ID and associated scholarship, is stored in a relational database for future reference and management.

5. **Error Handling:** In case of a failure during the channel creation process (e.g., invalid data or Discord API issues), the system will return a detailed error message to guide troubleshooting.

2.5.3 Benefits for Students

The integration provides several benefits for students:

- **Enhanced Interaction:** Students can directly interact with their peers and gain insights about the scholarship from Alumni.
- **Real-time Communication:** With real-time messaging, students can get immediate answers to their questions and participate in ongoing discussions.
- **Community Building:** The dedicated channels foster a sense of community and encourage collaboration among applicants, leading to stronger connections.

In conclusion, the Discord API integration offers a valuable tool for improving the scholarship application process. It creates an online space where students can exchange information, network with each other, and receive real-time support. This not only enriches the scholarship experience but also helps students build connections that can benefit them in their academic and professional journeys.

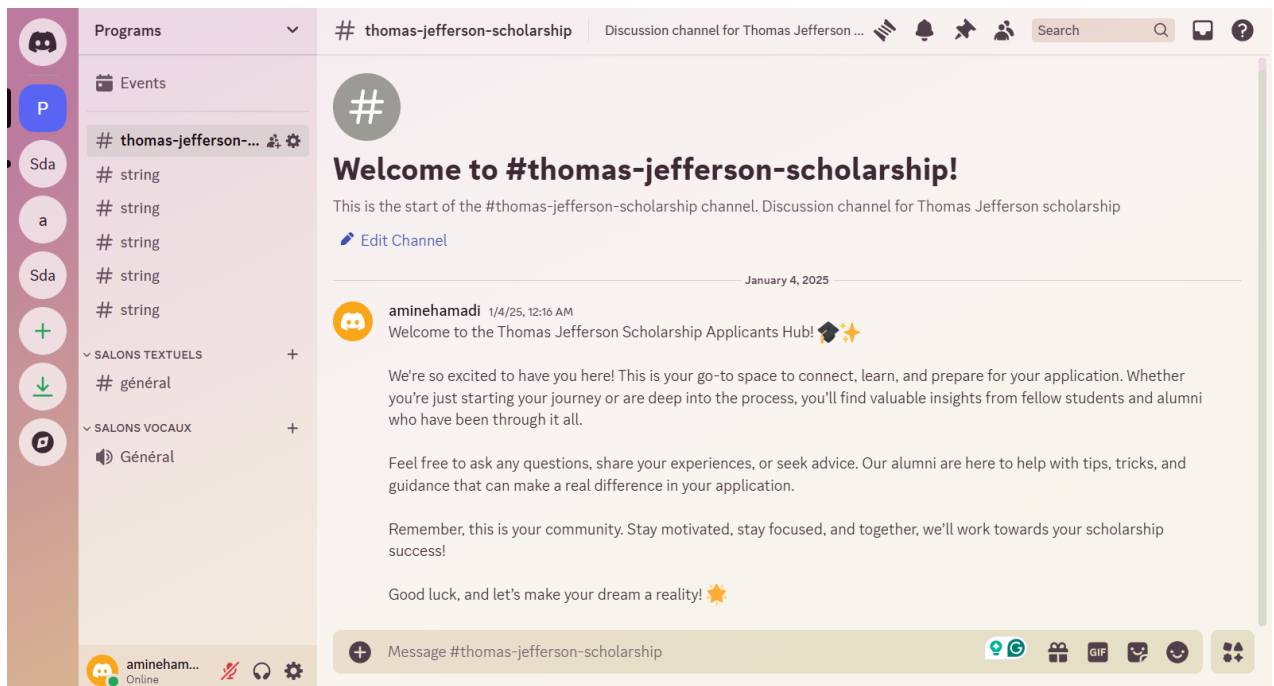


Figure 2.6: Discord channel for Thomas Jefferson Scholarship

2.6 Security Measures Implementation

The system incorporates multiple layers of security to protect sensitive information, including secure password storage, token-based authentication, and user session management.

2.6.1 Password Hashing

User passwords are hashed using the `bcrypt` algorithm before being stored, ensuring that even if the database is compromised, passwords remain secure. The `verify_password` function checks the inputted password against the stored hash during login.

2.6.2 Token-Based Authentication

Authentication is handled with **JWT (JSON Web Tokens)**. Upon login, users receive an access token and a refresh token. The access token expires after a short time (30 minutes), while the refresh token, stored in a secure HTTP-only cookie, lasts 15 days. The refresh token can be used to obtain a new access token, improving security and user experience.

2.6.3 Role-based access control

Role-based access control (RBAC) is employed to manage user permissions. The `get_current_user` function decodes the access token to authenticate users and ensure they have the necessary permissions. Invalid or missing tokens result in an HTTP 401 Unauthorized error.

2.6.4 Token Refresh and Revocation

The system allows users to refresh their access token using the refresh token. If the refresh token is missing or invalid, an error is raised. To prevent token hijacking, tokens are blacklisted upon logout, ensuring they can't be reused.

2.6.5 Password Reset and Update

Users can reset their passwords by providing a valid access token, after which the new password is securely hashed. For password updates, users must authenticate with their current password, and any mismatch raises a validation error.

In conclusion, the system ensures robust security through secure password handling, token management, and session control, safeguarding user data and preventing unauthorized access.

Chapter 3

Conclusion and Potential Improvements

3.1 Conclusion

In this project, we have developed a comprehensive system to facilitate scholarship applications and foster networking among students. The system integrates various components, including user management, secure authentication, a database to store essential data, and a unique feature that integrates with the Discord API. This integration enables students to engage in real-time discussions, ask questions, and network with their peers, significantly enhancing the scholarship experience.

The system is designed to ensure security, scalability, and ease of use, offering a seamless application process for students while maintaining robust data management practices. With the implementation of token-based authentication, password hashing, and secure token management, the system ensures that user data remains protected from unauthorized access. Furthermore, the integration with the Discord API provides a dynamic communication platform for students, facilitating real-time interaction and knowledge sharing.

In conclusion, this system effectively meets the core objectives of streamlining the scholarship application process while fostering a collaborative environment for students. By leveraging modern technologies such as FastAPI, PostgreSQL, and Discord's API, the project provides a scalable solution for future use and expansion.

3.2 Potential Improvements

While the system works effectively for the current use case, there are several areas where improvements can be made to enhance its functionality and user experience. Some potential improvements include:

1. **User-Friendly Website** Develop an intuitive, responsive website with easy navigation, providing students with a seamless experience. The website should offer easy access to scholarships, program details, tips, and reviews, ensuring that students can quickly find relevant opportunities and information.
2. **Automated Notifications** Implement an automated notification system to inform students about new opportunities, updates on their applications, or responses from

peers. Notifications can be delivered via email, in-app messages, or integrated Discord channels, ensuring students stay up-to-date.

3. **Gamification Elements** Add badges or points for engagement, such as writing reviews, sharing tips, or participating in discussions. This gamification approach will motivate users to actively engage with the platform and increase the overall user retention rate.

These improvements could help further strengthen the system's functionality, user engagement, and scalability, ensuring that it can accommodate future needs and user demands.

3.3 References

Here are the key references used in this project:

- FastAPI Documentation. <https://fastapi.tiangolo.com/>
- Discord API Documentation. <https://discord.com/developers/docs/intro>
- PostgreSQL Documentation. <https://www.postgresql.org/docs/>
- OAuth2 Security Protocol. <https://oauth.net/2/>
- JWT Authentication. <https://jwt.io/introduction/>
- bcrypt Hashing Algorithm. <https://en.wikipedia.org/wiki/Bcrypt>
- Pydantic Documentation. <https://pydantic-docs.helpmanual.io/>