CSS 418 – APPLICATION SECURITY

Documentation of GROUP 6 Course Registration Web App

Group Members

1. Francis Gideon Oghie - 2019/1/75540CS
2. Nelson Deji Ayo - 2019/1/75276CS
3. Joseph Itopa Raji - 2019/1/76392CS
4. Francis Kuje Anzabla - 2019/1/75856CS
5. Ibrahim Uthman Suleiman - 2019/1/76516CS
6. Adeyinka Rabiu - 2021/2/80942CS

Introduction

A web application (web app) is an application program that is stored on a remote server and delivered over the internet through a browser interface. This comprehensive guide details how to navigate the Cybersecurity Course Registration web app. This secure platform allows you to register for courses offered at your level (100-500) within the program. It's also secure as it does not allow registration from non-cybersecurity students.

Tech Stack

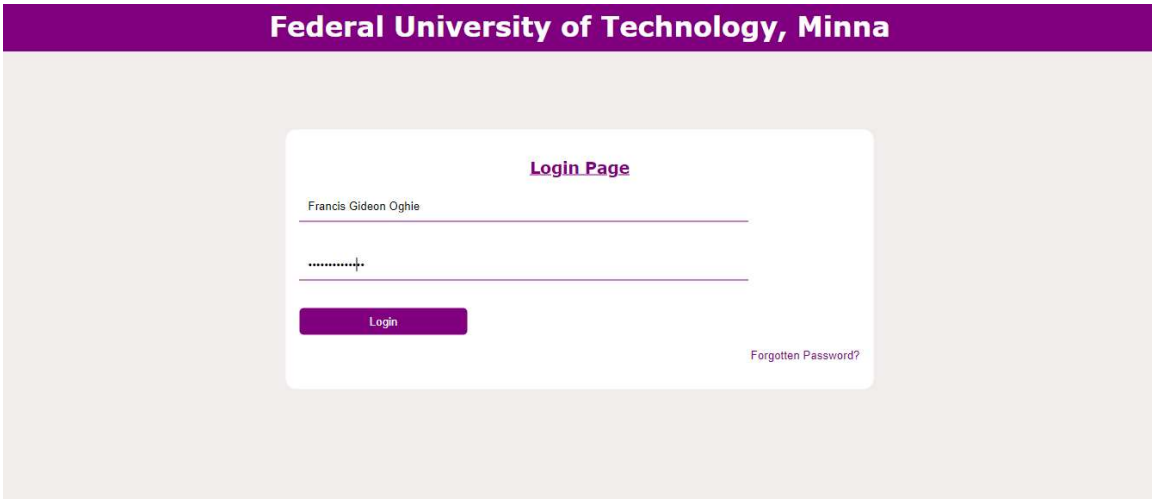
- HTML - Hypertext Markup Language
- CSS - Cascading Stylesheet
- JavaScript
- Firebase database
- Animation on Scroll(AOS) library

The site was hosted live on https://www.courseregapp.netlify.app

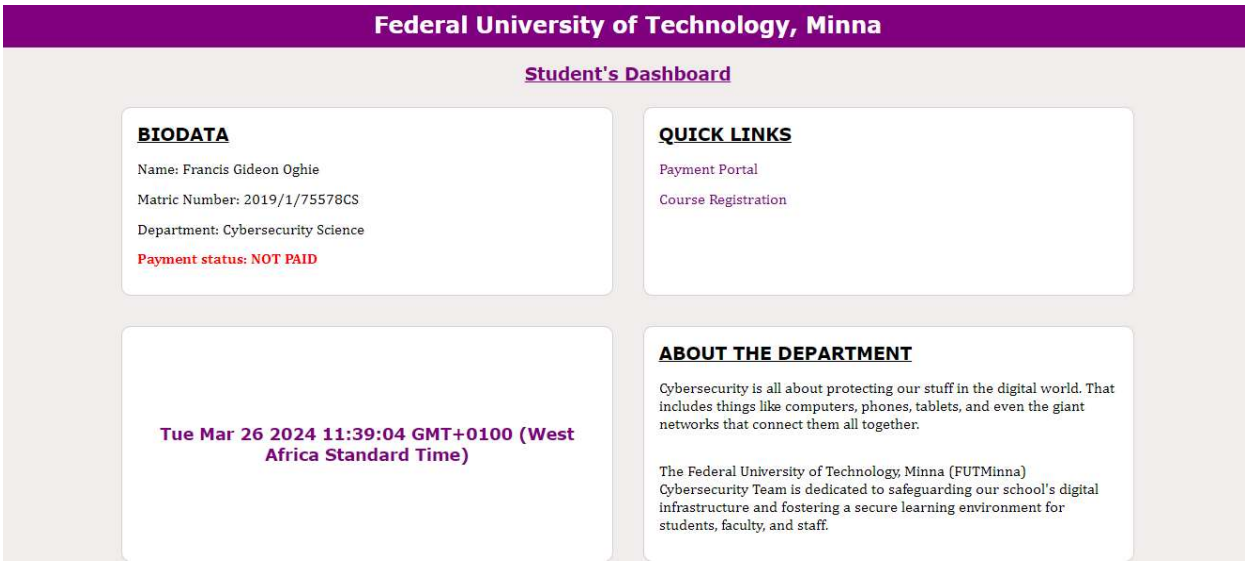
FRONTEND

Course Registration Process

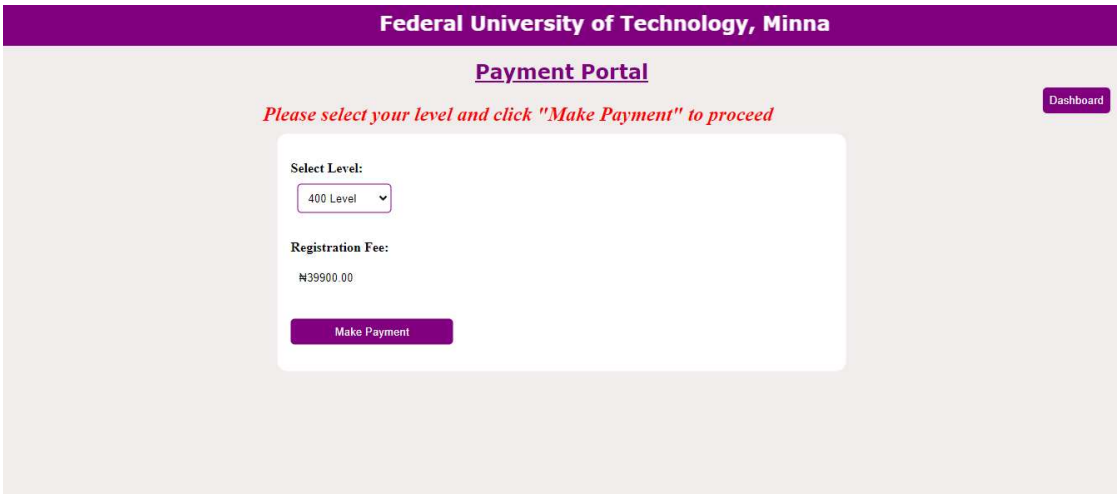
1. Access the Web App: Using a web browser, navigate to the official university website and locate the link to the Cybersecurity Coursen Registration web app. Proceed to enter your full name, and password. Your password is your matric number. Click the "Login" button to submit your information and be redirected to the dashboard.



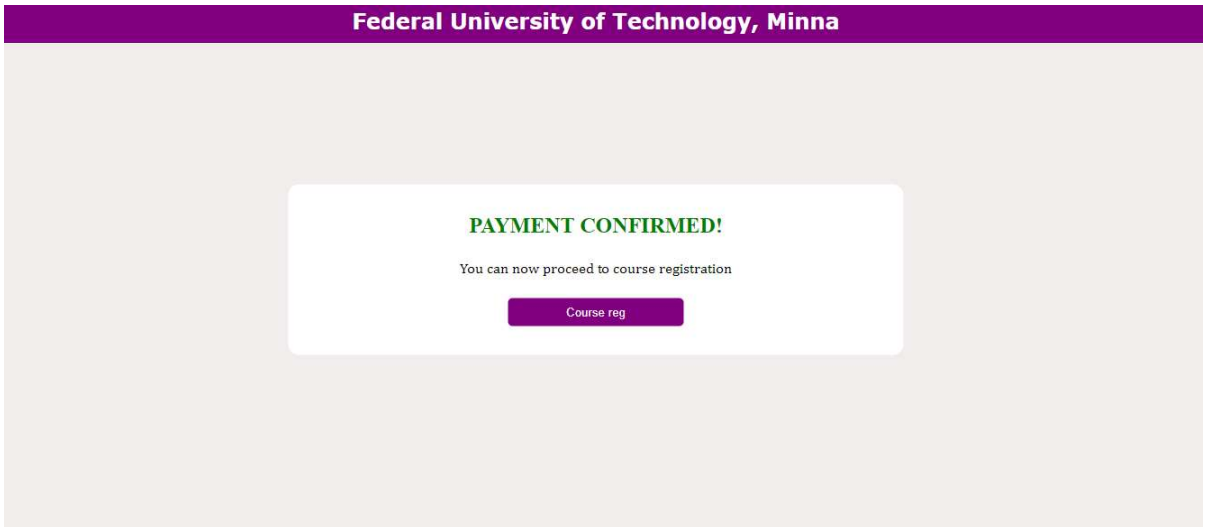
2. Dashboard (before payment): The dashboard contains relevant information about the candidate such as biodata, level, department and a dynamic payment status. It also contains information about the department as well as quick links to payment portal and Course registration.



3. Payment Gateway: To make payment of registration fee, you will be directed to a secure payment gateway. Follow the on-screen instructions to complete your school fee payment. You have to select your current level and then the school fees amount allocated to that level will be displayed.



4. Payment Confirmation: Once successful, a confirmation message will appear within the payment gateway. And you'll be directed to the course registration section.

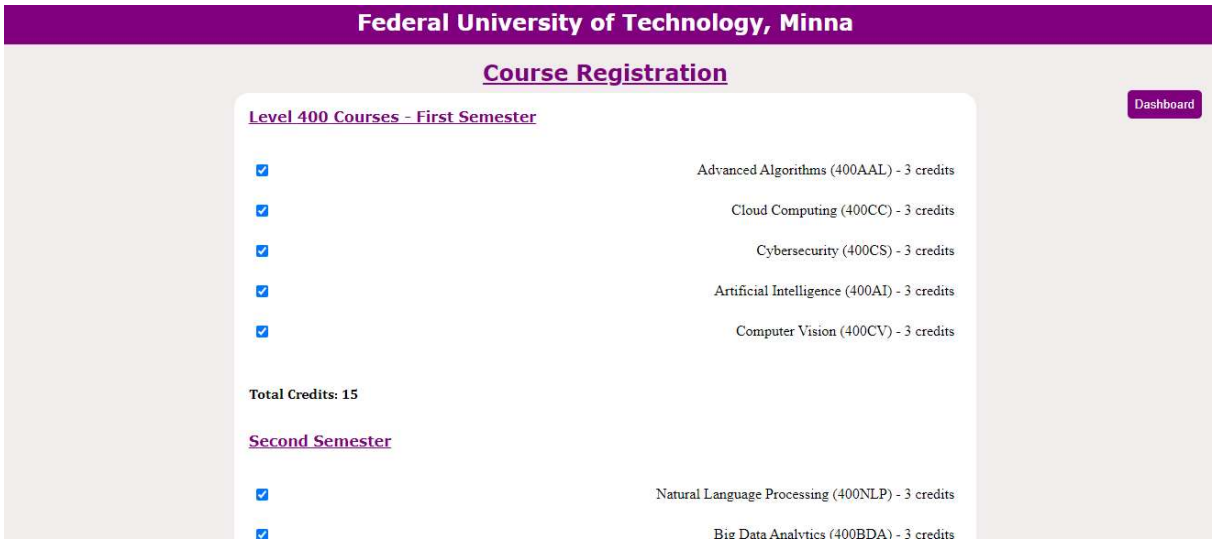


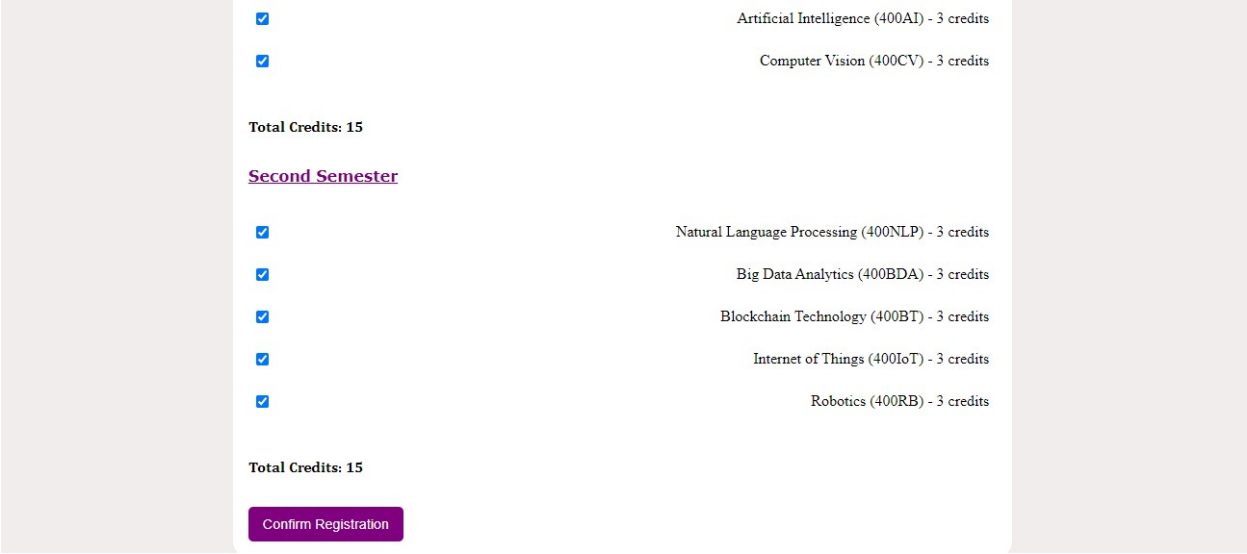
5. Course Selection and Viewing

1. ***Available Courses:*** Upon returning from the payment gateway, the web app will display a list of courses offered at your registered level (100-500).
2. ***Course List:*** This list will include details such as:

- ○Course Code
- ○ Course Name
- ○ Course Credit Unit

3. ***Select Courses:*** Click the checkboxes next to the courses you wish to register for. You can select multiple courses.
4. ***Review Selection:*** Click the "Confirm Registration" button to confirm your selections.





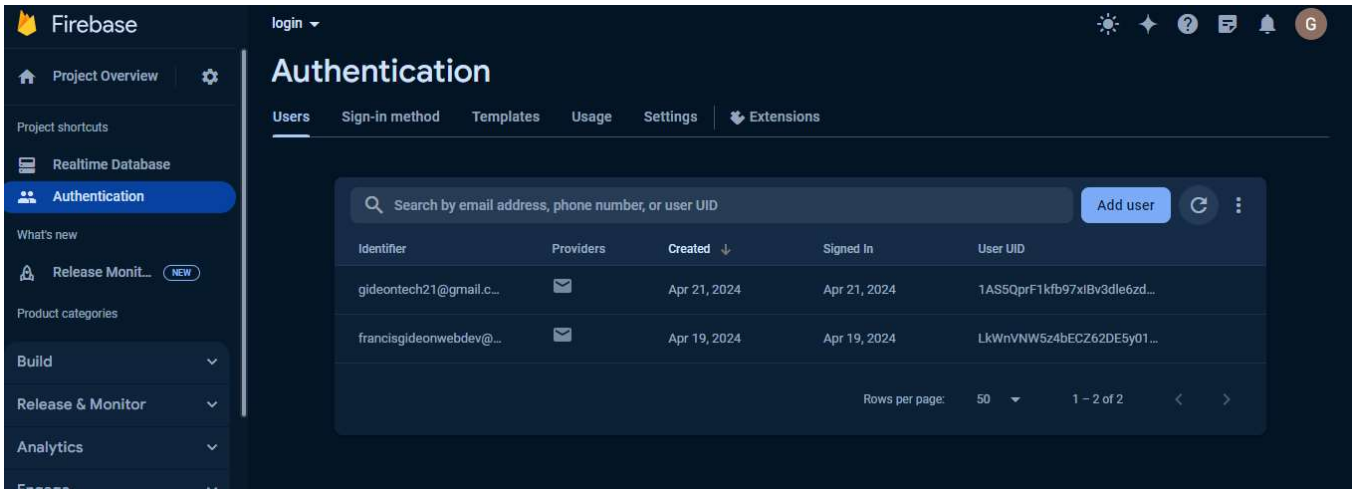
6. Registration Complete: Here you'll be notified that course registration is successful. The courses you selected will also be displayed on the screen with a congratulatory message.

7. Dashboard (after payment): After payment, the level and payment status of the student is updated on the dashboard.

BACKEND

The backend of our course registration web app enables authentication, and database storage as well as threat modeling and the necessary security policies to ensure the safety of user personal credentials

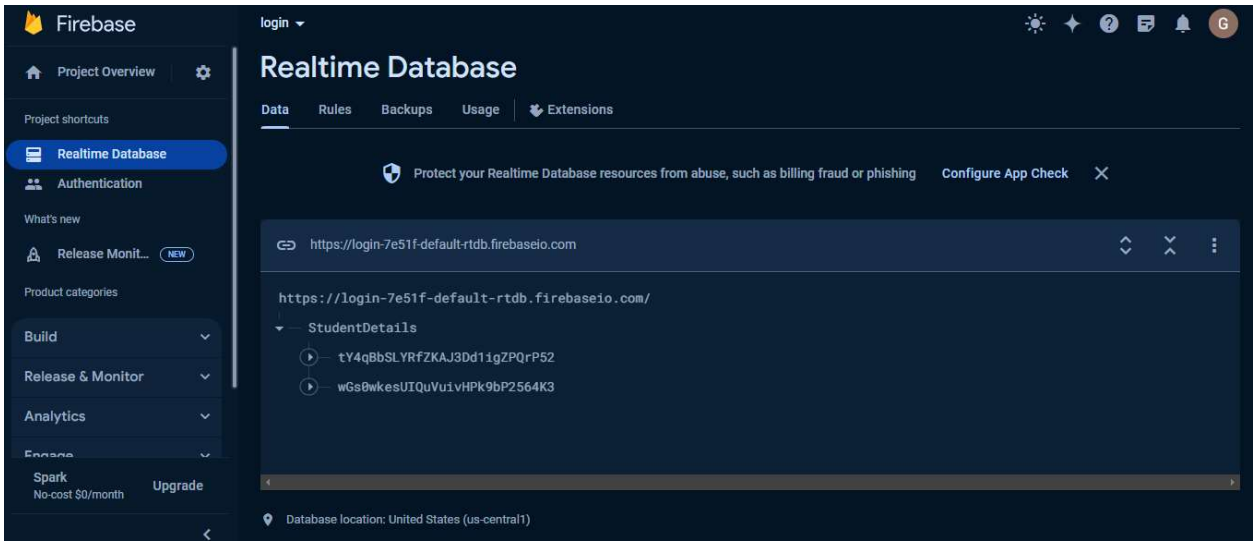
Authentication



The database stores the user input in the backend authentication section to check the validity of a user before granting access to the dashboard.

Database

Using the firebase realtime database feature, the user profile and input is stored on the database in real time. This realtime database ensures that changes are reflected immediately on the backend.



Threat Modeling

We did a thorough threat model on potential threats and security attacks that could adversely affect our web application and one that stood out was the Session Hijacking attack. And to prevent this, we implemented a unique session id for every user session on the browser.

Security Policies

We implemented the following security policies:

- User authentication before access to database
- Threat modeling to prevent various attacks such as session hijacking
- User input sanitization

- User password abstraction on the database in the case of a potential database breach
- Password hashing with SWYFT algorithm of 64-bits