

COMP-4990: Project Management

Smart Campus

Better Students' Life on Campus

Prepared by:

Jim Lin

Dawei Zhang

Chengyue Wei

Elias Tokola

Table of Contents

1. Background

- 1.1 Introduction
- 1.2 Digital transformation
- 1.3 Privacy
- 1.4 Work breakdown structure

2. Design

- 2.1 Technology stack
- 2.2 State management
- 2.3 UI design
- 2.4 MySQL database design

3. Project Management

- 3.1 Source control
- 3.2 Activities

4. Implementation

- 4.1 Home page
- 4.2 Courses page
- 4.3 Guidance page
- 4.4 Events page
- 4.5 Login page
- 4.6 Register page

5. Future Plans

6. Installation

- 6.1 React Js
- 6.2 Node Js
- 6.3 Database/XAMPP

7. Setup

- 7.1 Front end
- 7.2 Back end
- 7.3 Database

1. Background

1.1 Introduction

The SmartCampus project is an innovative website application designed to provide comprehensive university information and valuable insights from the community for new students. As they embark on their academic journey, this resource aims to ease the transition by offering a wealth of information about the campus, its resources, and the vibrant community they are joining. The project's goal is to empower new students with the knowledge they need to navigate their new surroundings and make informed decisions about their academic and social lives.

By offering detailed information on various courses and programs, the SmartCampus project allows students to explore their academic options and tailor their university experience to their individual needs and interests. The platform also enables students to gain insights into the courses they may find intriguing or beneficial, based on feedback and recommendations from fellow students and alumni.

Moreover, the SmartCampus project fosters a sense of camaraderie and support by connecting new students with the broader university community. Through forums, chat groups, and social events, incoming students can interact with their peers, faculty, and alumni, fostering a sense of belonging and ensuring they feel welcomed and supported throughout their academic journey.

In summary, the SmartCampus project serves as a valuable resource for new students by providing vital information, facilitating connections, and promoting a sense of community. With this platform, students can confidently navigate their university experience and make the most of their academic pursuits.

1.2 Digital Transformation

Digital transformation involves the integration of digital technology into all areas of an organization, including academic services, which is the focus of our project.

Our project aims to simplify the process of accessing course information for students. Additionally, we aim to provide an even better service to students in the form of course recommendations by the community.

Overall, our project aims to transform the academic services sector by offering a streamlined and personalized approach to course selection that simplifies the process for students and enhances their academic experience.

1.3 Privacy

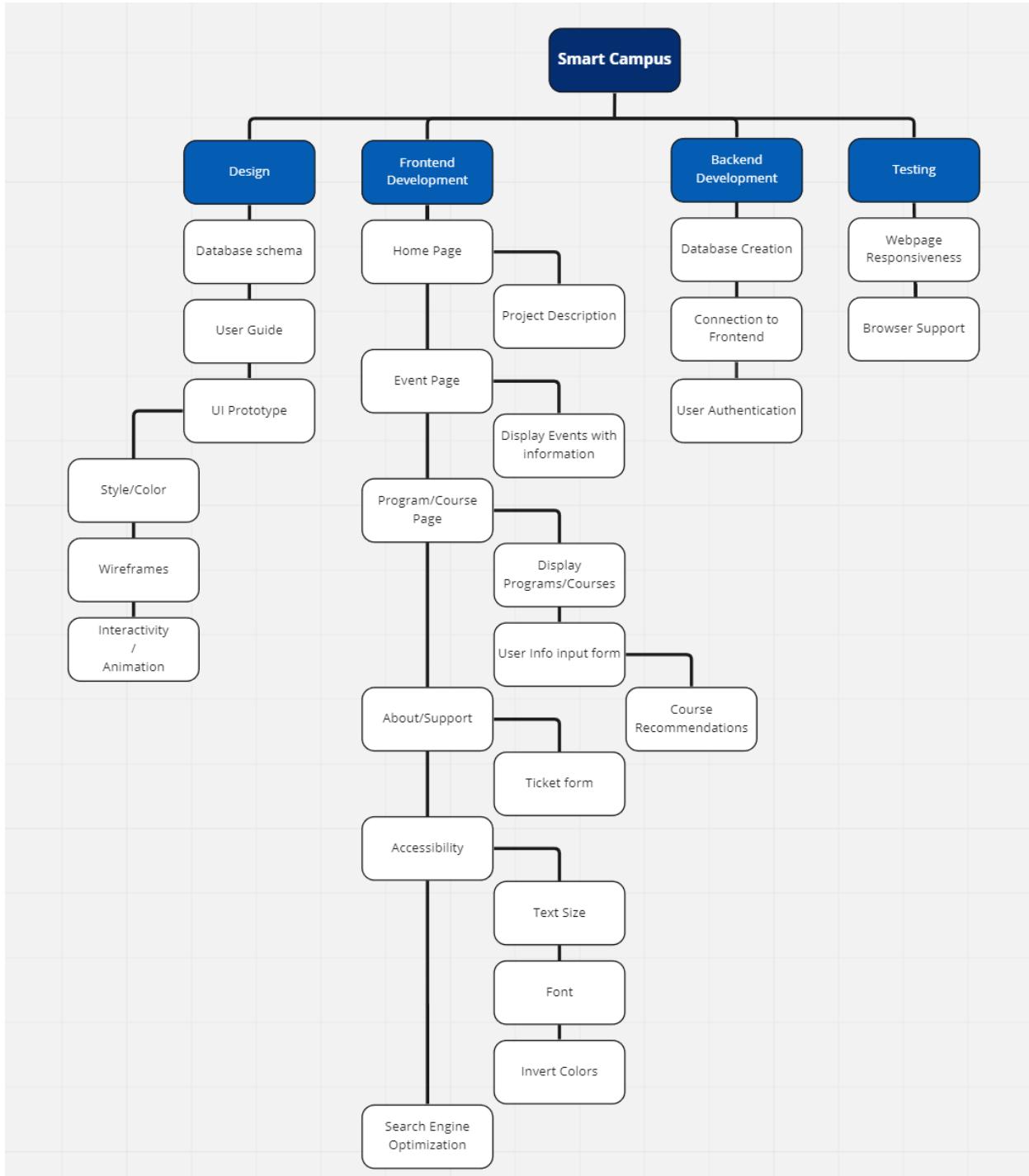
We recognize that privacy is a crucial aspect of the modern internet landscape, and protecting users' personal data has become more important than ever before. In an age where data breaches and identity theft are increasingly common, ensuring the privacy and security of personal information is a top priority. Unwanted exposure of sensitive data can lead to various negative consequences, including financial losses, emotional distress, and damage to one's reputation.

In our SmartCampus project, we are committed to safeguarding the privacy of our users by implementing strict data handling policies and practices. We understand that the less personal information we collect, the lower the risk of it being misused or compromised. Therefore, our platform will not request any personal details or data that could potentially infringe on users' privacy.

In addition to our proactive efforts to protect user privacy, we will maintain a transparent relationship with our users by providing clear and concise privacy policies. This way, users can understand how their data is being used and feel confident in their decision to engage with our platform.

At last, our SmartCampus project is dedicated to preserving the privacy and security of our users. By minimizing data collection, implementing robust security measures, and maintaining transparency, we aim to create a safe and trustworthy environment for students to access valuable university information and connect with their community without compromising their privacy.

1.4 Work Breakdown Structure



2. Design

2.1 Technology stack

React is an open-source JavaScript library developed and maintained by Facebook, primarily used for building user interfaces (UI) in web applications. React embraces the concept of reusable UI components, which allows developers to create modular and maintainable codebases. React's popularity can be attributed to its high-performance rendering capabilities, achieved through a virtual DOM (Document Object Model) that optimizes the updating process of the actual DOM. This approach minimizes the number of costly DOM manipulations, resulting in a faster and more efficient rendering of web applications. Additionally, React's one-way data flow and the integration of the Flux or Redux architecture contribute to more predictable and easier-to-debug applications.

Node.js is an open-source, cross-platform JavaScript runtime environment that executes JavaScript code outside the browser. Built on the V8 JavaScript engine from Google Chrome, Node.js allows developers to use JavaScript for server-side scripting, enabling them to build scalable and high-performance web applications.

JSON Web Token (JWT) is a compact, URL-safe means of representing claims to be transferred between two parties. JWTs are often used for authentication and authorization purposes in web applications. A JWT consists of three parts: a header, a payload, and a signature. The header typically contains information about the signing algorithm, while the payload includes the claims, such as user information or permissions. The signature is used to verify the token's integrity and ensure that it has not been tampered with. JWTs are stateless, meaning that they can be independently verified without requiring server-side storage. This property makes JWTs a popular choice for implementing secure authentication and authorization in modern web applications, particularly those built with microservices or serverless architectures.

Express is a lightweight, flexible, and unopinionated web application framework for Node.js that simplifies the process of building web applications and APIs. Express is designed to make it easy for developers to handle HTTP requests and responses, manage middleware, and create routes for different application endpoints. It provides a minimal and extensible interface, allowing developers to use plugins and middleware to add specific functionality as needed. Express also supports a wide range of template engines, such as EJS, Pug, and Handlebars, making it easy to render dynamic HTML pages on the server side. With its straightforward setup and ease of use, Express has become one of the most popular choices for building web applications with Node.js.

2.2 State management

Introduction to JWT:

- JSON Web Tokens are a compact, URL-safe means of representing claims to be transferred between two parties.
- Widely used for secure and stateless authentication and authorization.

Benefits of JWT:

- Stateless: Reduces server load by not requiring sessions to be stored.
- Scalable: Easily scales with increasing number of users and services.
- Secure: Signed tokens ensure integrity and prevent tampering.

Token lifecycle:

- Issued upon successful login or registration.
- Client-side storage (e.g., localStorage or cookies).
- Token sent with each request to protected resources.
- Server-side validation of token authenticity.
- Token expiration and renewal mechanisms in place.

Enhancing security:

- Implement short token expiration times.
- Use refresh tokens for extended sessions.
- Implement secure token storage and transmission methods.

Implementation in our project:

- JWT for authentication: Securely manage login and register states.
- Tokens stored client-side: Reduces server-side storage requirements.
- Tokens include user data: Simplify user identification and authorization.

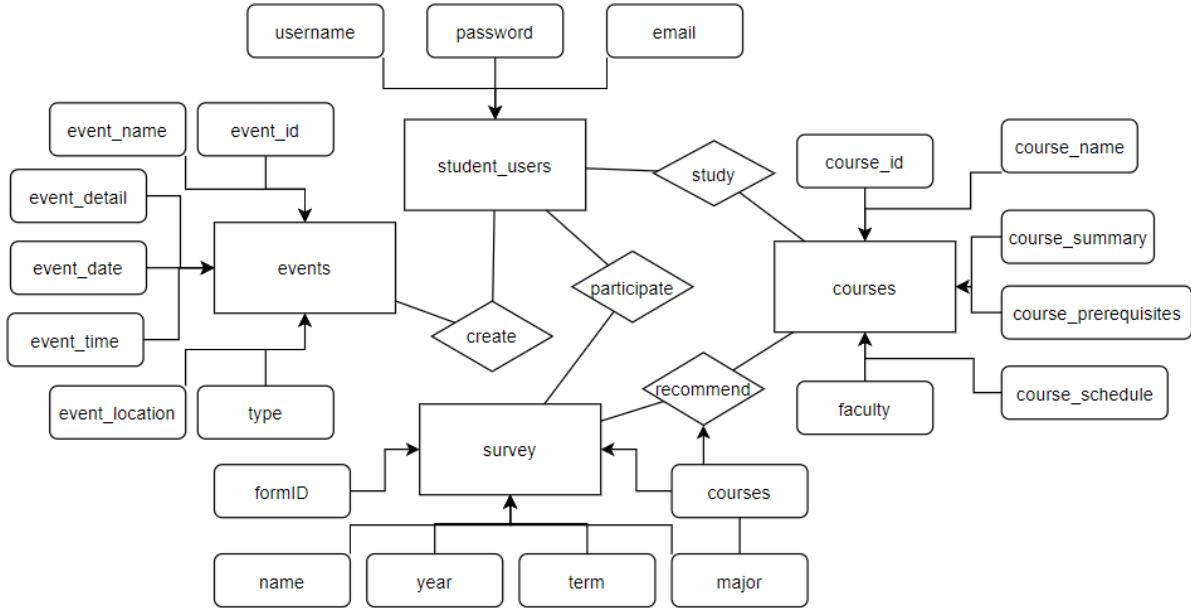
By utilizing JSON Web Tokens for state management, our project offers a secure, scalable, and efficient solution for handling user authentication and authorization.

2.3 UI design

To meet the user needs and goals, we identified the target audience, which is mainly students. We used consistent typography, color scheme, and layout. The design follows established design principles, such as alignment, hierarchy. These elements provide a clear and organized UI that is easy to understand and navigate.

We tested and received feedback during the UI design process and incorporated it into our design decisions. We used an iterative approach to continually refine and improve the design throughout the project.

2.4 MySQL database design



3. Project Management

3.1 Source control

Code part:

We decided to use GitHub to be our project version control system, because the Git Flow model suits our project development: each page and feature of this project are independent with each other, therefore, we can develop our pages and features in the develop branch, and merge it to the master branch once we've done that. And also, multiple versions could be produced when we were developing the project, and Git Flow can suit this better than other flows.

Communication Part:

We choose to use Discord as our communication platform, because all of us are familiar with this platform, it's easy to use, notification can be sent and received fast, functions are very useful (e.g. pin the message, upload files & screenshots),

3.2 Activities

2 Primary Phases:

Phase 1: UI Development (December 2022 - January 2023)

Jim: Designing the Course Page

Chengyue: Crafting the Registration & Login System

Dawei: Building the Home Page

Elias: Developing the Event Page & Navigation Bar

Phase 2: Database Integration & Digital Transformation Implementation (February 2023 - March 2023)

Chengyue & Elias: Establishing the database, backend infrastructure, and connecting the frontend through REST APIs.

Jim & Dawei: Populating the database with courses (covering approximately 75% of all courses) and linking the database to the Course Page.

Team: Collaboratively discussing the specifics of the digital transformation implementation and integrating it into the Guidance Page.

3.3 Documentation

1. Installation

a. client side:

- i. React.js (npm create-react-app .):
- ii. Axios

b. Server-side:

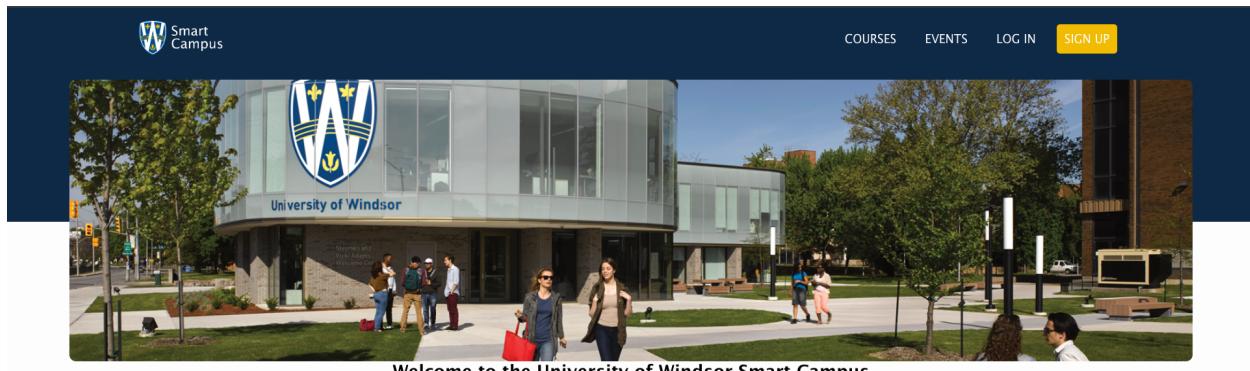
- i. Node.js (node server.js)

- ii. Express.js
 - iii. JWT
 - iv. Cors
 - v. BodyParser
- c. Database:
- i. MYSQL with phpMyAdmin
 - ii. XAMPP for local testing
1. Developer running commands
 - a. open XAMPP -> start Apache and MySQL -> Admin to MySQL -> get the database
 - b. go to server folder -> run command: node server.js
 - c. go to client folder -> run command: npm start

More details will be provided in the section down below

4. Implementation

4.1 Home page



Need to set a reminder for something?
Having a future plan?
Looking for a place to put all these things?
Check the [Events page](#)!

Looking for more information about the university?
Check the University [Website](#) !

4.2 Courses page

COURSES EVENTS LOGIN SIGN UP

University of Windsor

Filter Search courses... Guidance

AHSS

Education

Engineering

Graduate

Human Kinetics

ENG-1101 : Engineering 1

Description: Overview of the engineering profession: fields, career development, sustainability, health and safety, relation to society, business and entrepreneurship, equity, and Canada's Truth and Reconciliation process. Academic integrity, strategies for university success, academic regulations, engineering-extracurricular activities. Effective oral and written technical communication: informative and persuasive presentations; resumes and job search communications; technical writing and formatting; information gathering and analysis; research documentation and referencing; the use of visual tools such as graphs, figures, and tables; e-portfolios; and technical reports. (Open only to Engineering students.) (3 lecture hours and 1.5 tutorial hours weekly.)

More

The screenshot shows a university course search interface with a dark header bar. The header includes a 'Filter' button, a search bar with placeholder text 'Search courses...', a magnifying glass icon, and a 'Guidance' button.

The main content area displays three course cards:

- GENG-1101 : Engineering 1**
Overview of the engineering profession: fields, career development, sustainability, health and safety, relation to society, business and entrepreneurship, ethics, equity, and Canada's Truth and Reconciliation process. Academic integrity, strategies for university success, academic regulations, engineering-related extracurricular activities. Effective oral and written technical communication: informative and persuasive presentations; resumes and job search communications; technical writing and formatting; information gathering and analysis; research documentation and referencing; the use of visual tools such as graphs, figures, and tables; e-portfolios; and technical reports. (Open only to Engineering students.) (3 lecture hours and 1.5 tutorial hours weekly.) [More](#)
- GENG-1102 : Engineering Graphics**
Visualization techniques, graphical communication using sketching, descriptive geometry, and computer-aided design (CAD) for orthographic projection, pictorial drawings, dimensioning, section views, and auxiliary views. Reading engineering drawings. Engineering graphics e-portfolio and CAD project to develop visualization skills and task completion skills. (Open only to Engineering students.) (4.5 hours weekly.) [More](#)
- GENG-1110 : Engineering Mechanics I**
Statics of particles and rigid bodies; trusses, frames, machines; centroids and centres of gravity; friction. (3 lecture, 2 tutorial hours a week.) [More](#)

A modal window is open over the third card, displaying detailed information about **COMP-2540 : Data Structures and Algorithms**:

COMP-2540 : Data Structures and Algorithms

An introduction to the programming and analysis of linear and non-linear internal (main store) data structures and associated algorithms. Topics include the formal notion of an algorithm, elementary time and space complexity; linear lists (such as stacks, queues, linked structures.); non-linear lists (trees, binary trees); recursion; sorting techniques (such as heap sort, quick sort, merge sort, shell sort); searching techniques (such as binary search, binary search trees, red-black trees, hashing.); algorithm design paradigms (such as divide-and-conquer, dynamic programming, greedy algorithms); and applications.

Prerequisites: COMP-1000, COMP-1410
Terms: Fall/Winter/Summer
Faculty: Computer Science

[Back](#) [More](#)

Below the modal, another course card is partially visible: **COMP-2650 : Computer Architecture I**.

4.3 Guidance page

The screenshot shows the 'Academic Guidance' section of the Smart Campus website. At the top right, there are links for 'COURSES', 'EVENTS', 'LOGIN', and a yellow 'SIGN UP' button. Below these, the title 'Academic Guidance' is centered, followed by a sub-instruction: 'Here you can find some helpful resources and links for academic guidance.' A yellow button labeled 'Recommend your schedule' is positioned below the text. The main content area contains four blue rectangular boxes, each representing a student's academic information:

- Ame : Bachelor of computer science**
Fall 2022 : COMP-1000, COMP-1400, COMP-1047
- Bob : Bachelor of computer science**
Winter 2022 : COMP-2140, COMP-3150, COMP-2650, COMP-2750
- Christine : Bachelor of computer science**
Summer 2022 : COMP-2560, COMP-2650, COMP-2707
- Dylen : Bachelor of computer science**
Fall 2019 : COMP-3110, COMP-3150

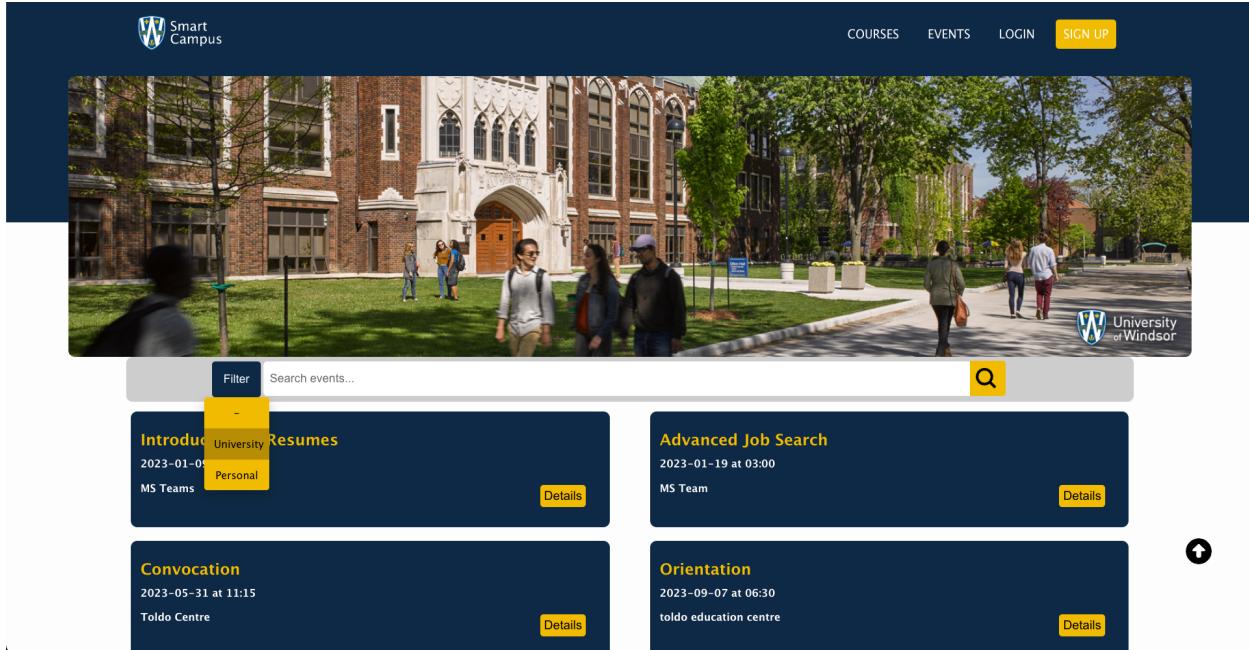
The screenshot shows a 'Survey' modal window overlaid on the Smart Campus website. The modal has a dark gray background and contains fields for user information:

- Name:
- Year:
- Term:
- Major:
- Courses:

At the bottom right of the modal is a dark blue 'Submit' button.

In the background, the 'Academic Guidance' section is visible with the same four student profiles as the first screenshot.

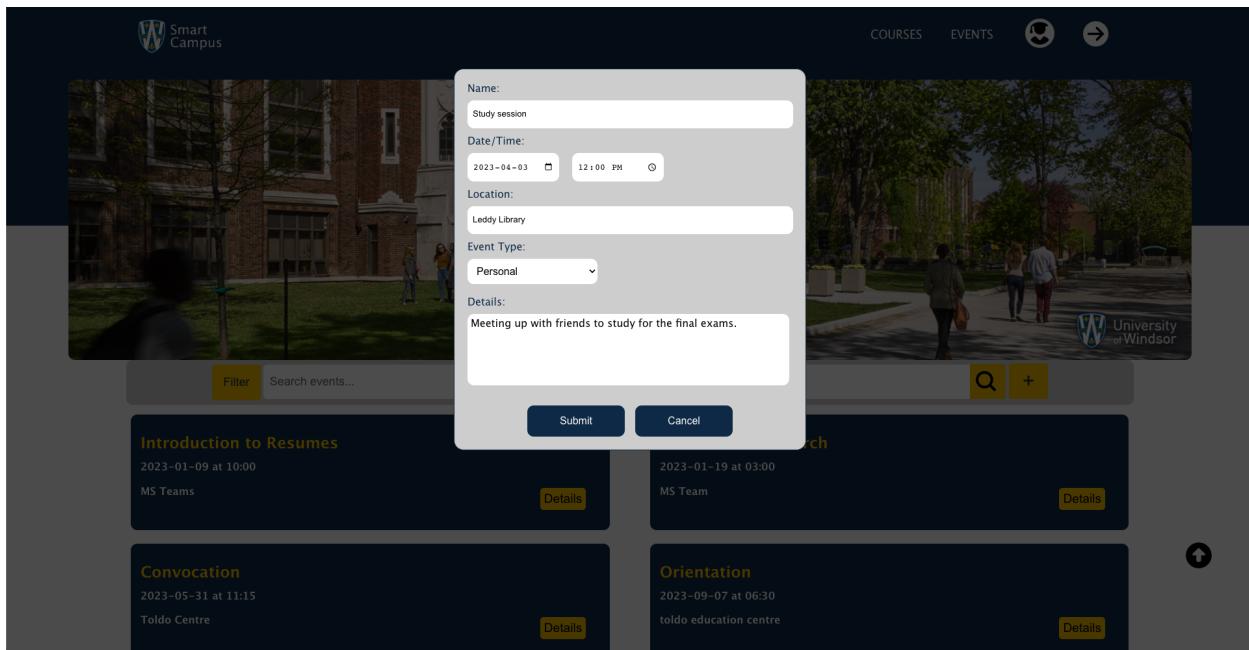
4.4 Events page



The top section shows the Smart Campus homepage with a banner image of students walking on a campus path. Navigation links include COURSES, EVENTS, LOGIN, and SIGN UP.

Events Grid:

- Introduction to Resumes**: 2023-01-09 at 10:00, MS Teams. Filtered by Personal.
- Advanced Job Search**: 2023-01-19 at 03:00, MS Team.
- Convocation**: 2023-05-31 at 11:15, Toldo Centre.
- Orientation**: 2023-09-07 at 06:30, toldo education centre.



A modal window is open for creating a new event:

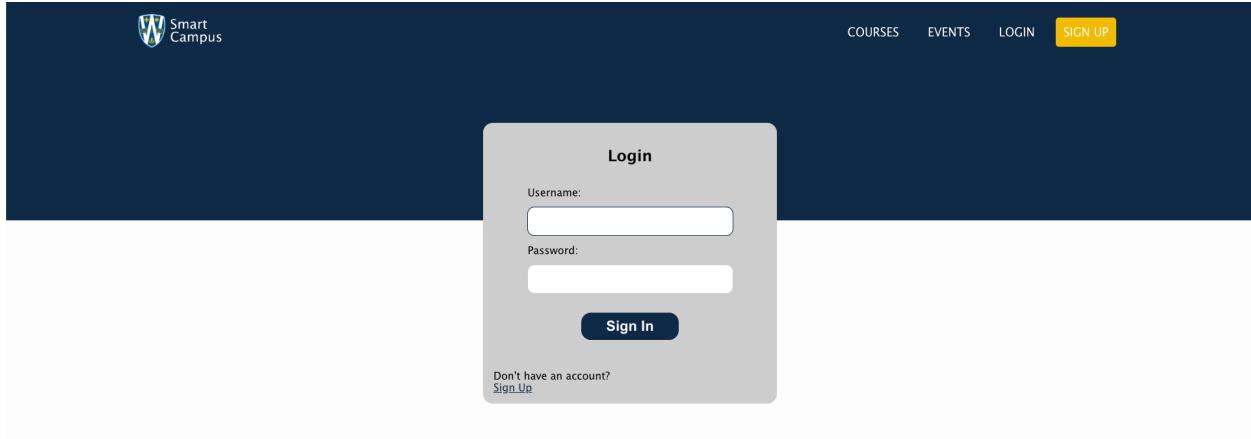
Event Details:

- Name: Study session
- Date/Time: 2023-04-03 12:00 PM
- Location: Leddy Library
- Event Type: Personal
- Description: Meeting up with friends to study for the final exams.

Buttons: Submit, Cancel.

The background shows the same events grid as the first screenshot.

4.5 Login page

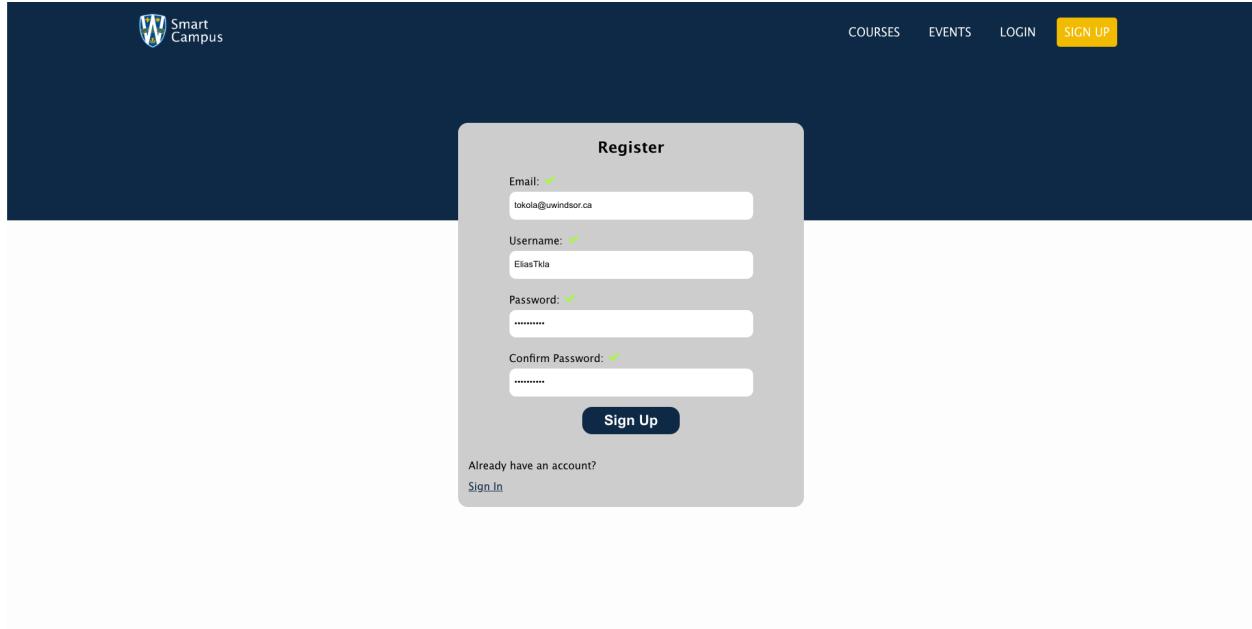


The screenshot shows the Smart Campus events page. The top section features a large, horizontal photograph of a university campus with students walking on a path in front of a red brick building with arched windows. Above the photo is a dark blue header bar with the "Smart Campus" logo, "COURSES", "EVENTS", and other navigation links. To the right of the header are icons for user profile, search, and navigation. Below the photo is a search bar with a placeholder "Search events..." and a magnifying glass icon. To the right of the search bar is a plus sign icon. The main content area displays four event cards in a grid:

- Introduction to Resumes**
2023-01-09 at 10:00
MS Teams [Details](#)
- Advanced Job Search**
2023-01-19 at 03:00
MS Team [Details](#)
- Convocation**
2023-05-31 at 11:15
Toldo Centre [Details](#)
- Orientation**
2023-09-07 at 06:30
toldo education centre [Details](#)

A small upward-pointing arrow icon is located in the top right corner of the event grid.

4.6 Register page



5. Future plans

- Implement UWindsor email authentication for login.
- Improve the courses & events page to provide more information and interactability (ratings, comments, attendance, etc.).
- Incorporate students' academic history/grades with an AI to provide detailed recommendations, study plans, etc.
- Broaden the scope of information provided within the website to allow for more capabilities within a smaller stream of access (maps, career opportunities).
- Allow students to easily connect and share thoughts and experiences on programs, courses, events, and more (individual/group chats).
- Notifications/emails through multiple devices (mobile, desktop, laptop, smartwatch).

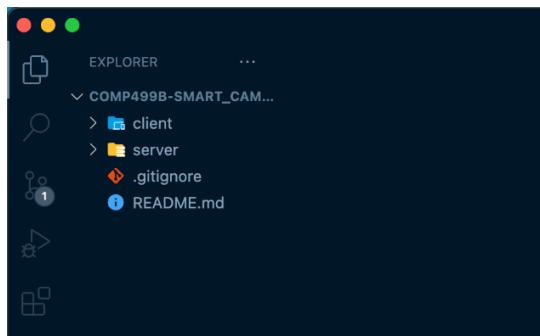
6. Installation

6.1 React Js

Clone the repository from the github at:github.com/KevinVWei/comp499b-smart_campus

```
-MacBook-Air ~ % git clone https://github.com/KevinVWei/comp499b-smart_campus.git
```

Open the project within your IDE, in this case it's VSCode.



Open a terminal and change directories to the Client folder.

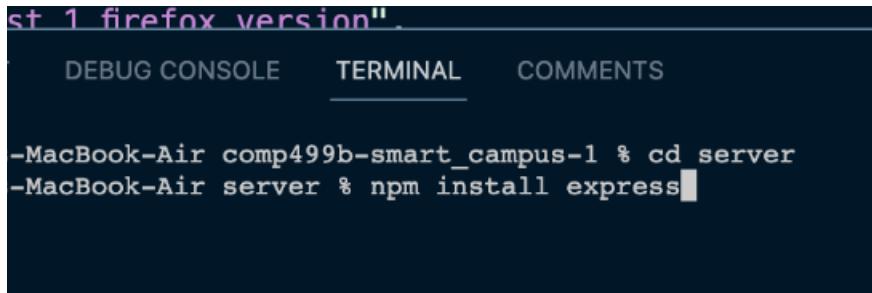
```
DEBUG CONSOLE TERMINAL COMMENTS

-MacBook-Air comp499b-smart_campus-1 % cd client
-MacBook-Air client %
```

1. Run ‘npm install’
2. Run ‘npm install axios’

6.2 Node Js

Open a terminal within the Server folder.



The screenshot shows a terminal window with three tabs: DEBUG CONSOLE, TERMINAL, and COMMENTS. The TERMINAL tab is selected. The command 'npm install express' is being typed into the terminal.

```
st 1 firefox version".
DEBUG CONSOLE    TERMINAL    COMMENTS

-MacBook-Air comp499b-smart_campus-1 % cd server
-MacBook-Air server % npm install express
```

Now you must download the dependencies with ‘npm install “dependency”’

1. Run ‘npm install express’
2. Run ‘ npm install bodyParser’
3. Run ‘ npm install cors’
4. Run ‘ npm install mysql’
5. Run ‘ npm install jsonwebtoken’
6. Run ‘npm install nodemon’ – not required but simplifies local testing

6.3 Database/XAMPP

To develop the project locally you require the use of MySQL database manager. We used XAMPP to create and run a local database instance.

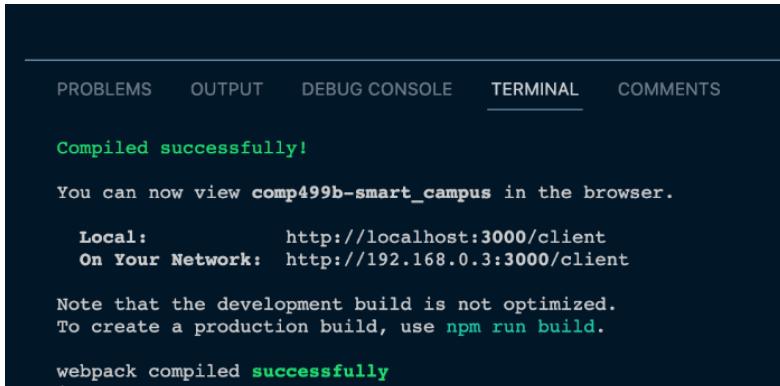
Download [XAMPP](#) and follow the instructions to install the application.

7. Setup/Run

7.1 Front end

Open a terminal within the client folder and you can run the client frontend with ‘npm start’, and the app will launch on a new browser page. The url of the page will likely be <http://localhost:3000/> unless there is something else running on that port within your machine.

The following should be displayed if the app starts successfully, stating the server is currently running on.



```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    COMMENTS

Compiled successfully!

You can now view comp499b-smart_campus in the browser.

Local:          http://localhost:3000/client
On Your Network:  http://192.168.0.3:3000/client

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully

```

7.2 Back end

Open a second terminal within the Server folder and you can run the server with ‘node server.js’, but to simplify the development cycle, use ‘nodemon --watch’. This will start the server and automatically rerun it when changes are made to the server.js file.

This should be displayed if the server starts successfully using ‘nodemon --watch’.



```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    COMMENTS

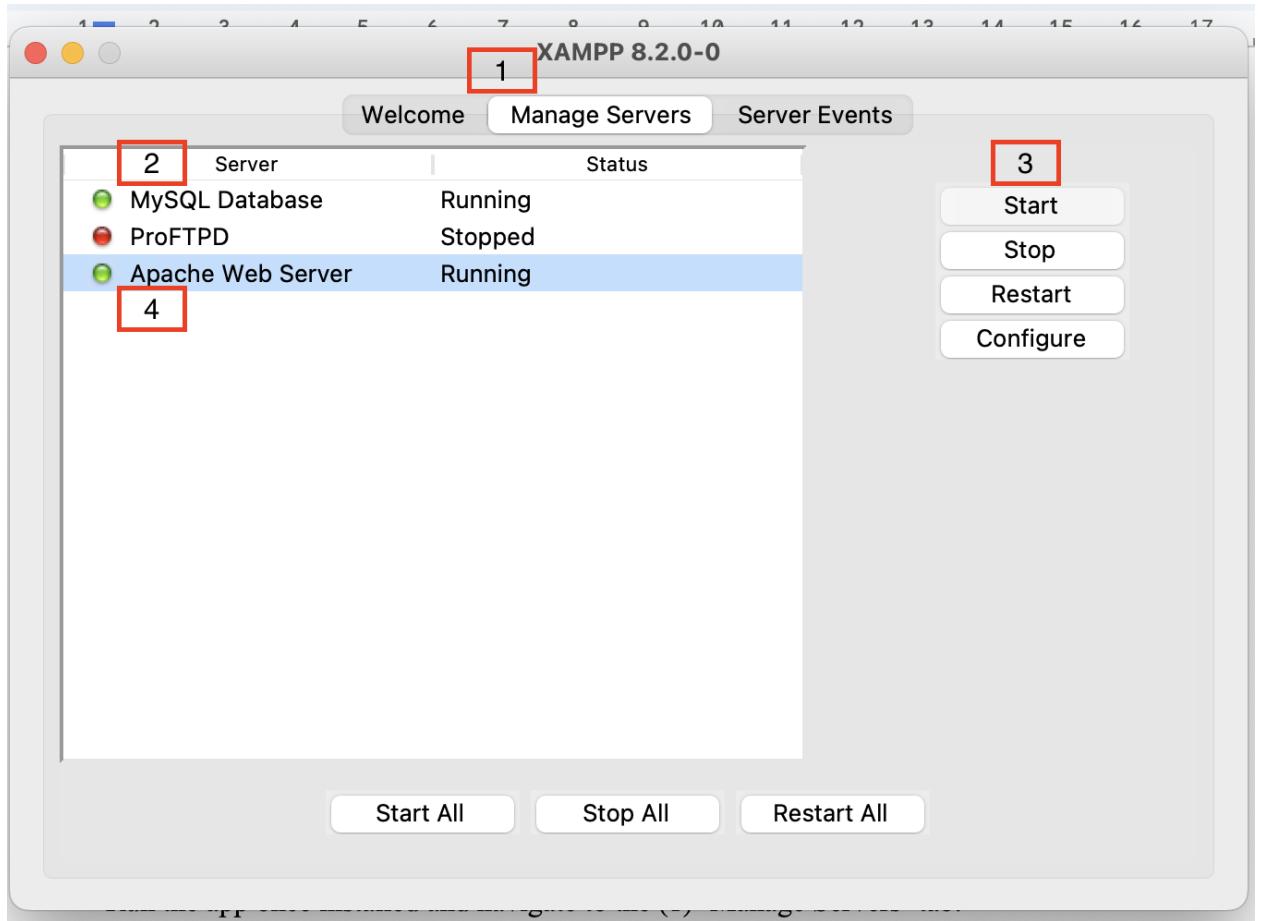
● eliastokola@Eliass-MacBook-Air comp499b-smart_campus % cd server
○ eliastokola@Eliass-MacBook-Air server % nodemon --watch
[nodemon] 2.0.20
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): ***!
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node server.js'
Server started on port 4000
■

```

7.3 Database

Run the XAMPP app and navigate to the ‘Manage Servers’ tab (1).

Then select the ‘MySQL Database’ (2) and start it (3). Do the same with the ‘Apache Web Server’ (4).



Now you can head to a browser and search the link <http://localhost:80/phpmyadmin>, to reach the phpMyAdmin database manager.

The screenshot shows the phpMyAdmin interface. The top navigation bar shows the URL "localhost/phpmyadmin". Below the header, there is a toolbar with various icons and a dropdown menu. The main content area has a left sidebar with a tree view of databases: "bookquest", "cm_4990", "information_schema", "mysql", "performance_schema", "phpmyadmin", and "test". The main panel displays the "General settings" section. It includes a dropdown for "Server connection collation" set to "utf8mb4_unicode_ci" and a "More settings" link. Below this is the "Appearance settings" section. The entire interface is in light mode.

Once you have reached the site, create a new database with the following name or any other name you see fit.

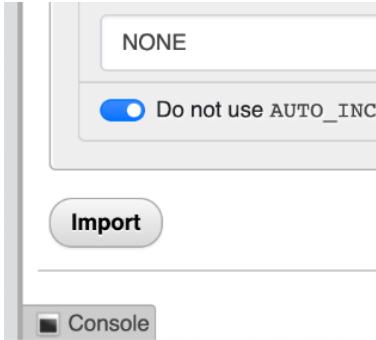
The screenshot shows the 'Databases' section of the phpMyAdmin interface. On the left, there's a sidebar with a 'New' button and a list of existing databases: bookquest, cm_4990, information_schema, mysql, performance_schema, and phpmyadmin. The main area is titled 'Databases' and features a 'Create database' form. In the 'Name' field, 'testdb' is entered, and the 'Character set' dropdown is set to 'utf8mb4_general_ci'. A 'Create' button is visible at the bottom right of the form.

Once the database is created, navigate to the database and head to the import panel from the button named import.

The screenshot shows the 'Import' panel for the 'testdb' database. The title bar indicates 'Database: testdb'. The main content area is titled 'Importing into the database "testdb"'. It contains a 'File to import:' section with instructions: 'File may be compressed (gzip, bzip2, zip) or uncompressed. A compressed file's name must end in .[format].[compression]. Example: .sql.zip'. Below this is a 'Browse your computer: (Max: 40MiB)' button and two buttons: 'Choose File' and 'No file chosen'.

Here you will have to upload the Database.sql file that is within the project folder, and click the import button on the bottom of the page without changing any other field.

The screenshot shows the 'Import' panel for the 'current server'. The title bar indicates 'Server: localhost'. The main content area is titled 'Importing into the current server'. It contains a 'File to import:' section with instructions: 'File may be compressed (gzip, bzip2, zip) or uncompressed. A compressed file's name must end in .[format].[compression]. Example: .sql.zip'. Below this is a 'Browse your computer: (Max: 40MiB)' button and two buttons: 'Choose File' and 'Database.sql'. Further down, it says 'You may also drag and drop a file on any page.' and 'Character set of the file:' with 'utf-8' selected.



This should be displayed if the file was imported successfully and the table along with their records are created. Now the database is ready to be accessible from the server for the frontend.

```

-- phpMyAdmin SQL Dump
-- version 5.2.0
-- https://www.phpmyadmin.net/
-- Host: localhost
-- Generation Time: Mar 24, 2023 at 11:15 PM
-- Server version: 10.4.27-MariaDB
-- PHP Version: 8.2.0
SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
START TRANSACTION;
SET time_zone = "+00:00";
CREATE TABLE `courses` (
    `id` int(11) NOT NULL,
    `name` varchar(255) NOT NULL,
    `description` text NOT NULL,
    `start_date` date NOT NULL,
    `end_date` date NOT NULL,
    `instructor_id` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
CREATE TABLE `events` (
    `id` int(11) NOT NULL,
    `name` varchar(255) NOT NULL,
    `description` text NOT NULL,
    `start_time` datetime NOT NULL,
    `end_time` datetime NOT NULL,
    `location` varchar(255) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
CREATE TABLE `student_users` (
    `id` int(11) NOT NULL,
    `first_name` varchar(255) NOT NULL,
    `last_name` varchar(255) NOT NULL,
    `email` varchar(255) NOT NULL,
    `password` varchar(255) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
CREATE TABLE `survey` (
    `id` int(11) NOT NULL,
    `question` text NOT NULL,
    `option1` varchar(255) NOT NULL,
    `option2` varchar(255) NOT NULL,
    `option3` varchar(255) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

```