# Contact Information

Development Director – Aditya Patel - <u>adityapatel@dal.ca</u> Technical Director – Madhulika Reddivari - MReddivari@dal.ca

# 1. Project Overview

Project Name: Design Your Degree

**Overview:** The design your degree is a web based academic tool that aims to provide an alternative option to students to plan their degree, customize it, and visualize their university degree progress. It enables students to use drag-and-drop courses across terms and years, select electives, and track completion status with an interactive UI. The tool enforces **prerequisite validation**, ensuring academic rules are followed when planning course sequences.

The system supports **year-tab navigation**, term-wise course organization, and filters for **core**, **elective**, **and option courses**. A role-based architecture enables students, advisors, and admins to interact with the platform according to their permissions. It adheres to **Dalhousie University's branding guidelines** for a consistent and accessible user interface. Key technologies include **React.js** for the front end, **Python** for the backend, and **MySQL** for the database integration.

At present, the following features are operational:

- **Dashboard for Academic advisors** is implemented on the frontend, but it is just visible. This can be changed as per the client requirements moving ahead and also show student data on request.
- Saving and retrieving data from database for the progress made and saved by students.
- **Security can be improved** with the help of hashing functionality. This will help the team to keep the data of the students secure.

The project is currently hosted on the outreach2.cs.dal.ca server, provided by Dalhousie University's Faculty of Computer Science. All source code is managed through GitLab with full version control. The system is stable, modular, and well-organized, offering a strong foundation for future teams to extend its functionality with minimal onboarding effort. Some files are intentionally left empty, serving as placeholders for features planned for future development.

# 2. Work Completed

# Summary of Work Done:

- o Implemented UI styling and components in accordance with Dalhousie University branding guidelines.
- o Developed secure login and signup functionality.
- Integrated drag-and-drop functionality for intuitive course selection and scheduling.
- Enabled navigation across academic years, allowing users to move and organize courses by term.
- o Added functionality for selecting free elective courses from an available list.
- o Built a program options module supporting the addition of minors, certificates, and honours designations.
- o Prerequisite logic is implemented at the backend but and also integrated into the course selection workflow.

# Deviations from Project Plan:

- User progress was planned to be saved in real time to the database but is currently stored via browser cookies due to time constraints.
- The entire degree plan was intended to be shown on a single scrollable page but was partially implemented using year-based tabs.

# 3. Codebase and Documentation

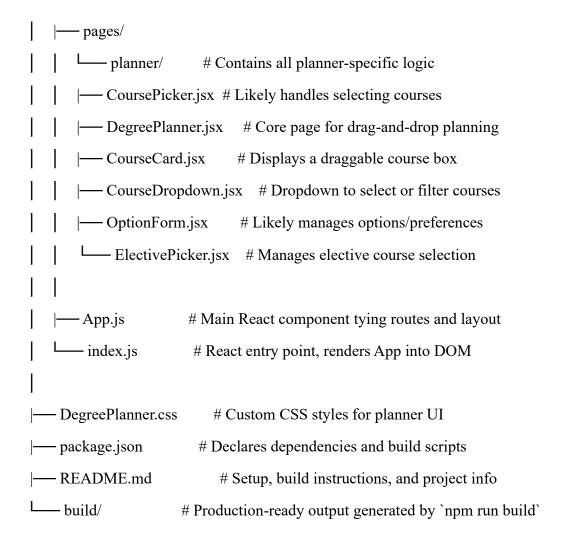
# 3.1 Overview of the code structure and key components.

Our project is built with a modular structure and clear separation between frontend and backend. The codebase is divided into three main directories: frontend, backend and database.

# 3.1.1 Frontend Folder Hierarchy Overview

Below is a structured representation of the frontend directory layout:

#### frontend/



At the top level, the public/ directory contains static assets, including the index.html file where the React application is mounted. The src/ directory contains all source code and is the core of the application. Within it, the pages/planner/ subdirectory is the most significant section, it contains the complete logic and layout for the interactive Degree Planner, including support for drag-and-drop, course selection, and term-based organization. Shared or reusable components are stored in the components/ directory.

The entry point of the application is defined in index.js, which renders the App.js component into the DOM. The App.js file ties together the page structure and manages routing or global layout if needed. Custom styles are managed through DegreePlanner.css, which contains specific rules for spacing, layout, and responsiveness of the planner interface.

# **Important Files to Know**

- DegreePlanner.jsx: Main component that renders the planner layout and handles dragand-drop logic.
- CourseCard.jsx: Defines how each draggable course appears.
- CourseDropdown.jsx, ElectivePicker.jsx, OptionForm.jsx: Handle course selection and user interaction.
- CoursePicker.jsx: Utility used to retrieve and display course options.

# **Technologies and Libraries Used**

- **React.js** for component-based UI development.
- @hello-pangea/dnd for drag-and-drop support.
- JavaScript (ES6+) as the primary language.
- CSS for layout and visual styling.
- **npm** for dependency and script management.

Build artifacts are generated in the build/ folder using npm run build, and this folder should never be edited manually. Dependencies and scripts are declared in package.json.

# 3.2 Link to the code repository and relevant documentation in Gitlab.

https://git.cs.dal.ca/courses/csci-x691/design-your-degree.git

# 3.3 Step-by-step instructions for setting up the development environment.

To set up the development environment for the **Design Your Degree** project, please follow the steps outlined below:

# 1. Clone the Repository: -

Clone the project repository from GitLab using the following command:

git clone https://git.cs.dal.ca/courses/csci-x691/design-your-degree.git

# 2. Navigate to the Project Root Directory

cd design-your-degree

#### 3. Install Frontend Dependencies

Navigate to the frontend directory and install all required Node.js packages:

cd frontend

# npm install

# 4. Install Backend Dependencies

Return to the root directory and then navigate to the backend folder. Install all required Python packages:

cd ../backend

pip install -r requirements.txt

# 5. Configure Environment Variables

Inside the backend directory, duplicate the .env.example file and rename it to .env. Update the environment variables in the .env file as needed (e.g., database connection strings, API keys).

#### 6. Start the Backend Server

From the backend directory, run python app.py and for MacOS it is python3 app.py

#### 7. Start the Frontend Server

Navigate back to the frontend directory and start the React development server:

cd ../frontend

npm start

# 3.4 Database Setup and Configuration

To set up the MySQL database for the Design Your Degree project on your local machine, follow the steps below:

# **Step 1: Create the Database**

Open MySQL Workbench or your terminal and run the following command to create the database:

CREATE DATABASE IF NOT EXISTS DegreePlanner;

USE DegreePlanner;

### Step 2: Run the main schema script

Navigate to the project's 'database' folder and run the 'design-your-degree.sql' script. This script will create the core tables such as:

- Programs

- Users
- Subjects
- Terms
- AcademicYears

#### Command:

mysql4 -u root -p DegreePlanner < database/design-your-degree.sql

# **Step 3: Run additional setup scripts**

# Run the following SQL scripts one by one to create other necessary tables:

1. Add Course Status Field:

mysql -u root -p DegreePlanner < database/add\_course\_status\_field.sql

2. Add Security Questions Table:

mysql -u root -p DegreePlanner < database/add security questions.sql

3. Create Prerequisites Table:

mysql -u root -p DegreePlanner < database/create prerequisites table.sql

# Step 4: (Optional) Seed initial data

If needed, run the Python seeding script to populate basic term data:

python database/seed terms.py

Note: Make sure your virtual environment is activated, and database credentials are correctly set in your `.env` file.

Your database is now fully set up on your local device and ready to be used with the backend and frontend servers. The project maintenance process is streamlined through Git, with the development branch designated for hosting updates and version control. Any changes or new features should be pushed to this branch and then redeployed as needed. For deployment, the build folder (generated using npm build) must be uploaded to the /home/project-dyd/design-your-degree directory on the server via FileZilla, which updates the hosted site at <a href="https://design-your-degree.research.cs.dal.ca/">https://design-your-degree.research.cs.dal.ca/</a>.

Although I initially encountered challenges while hosting, Chris Maxwell from the FCS Help

Desk provided significant guidance on the server structure and the deployment process. The actual directory being served is <u>/local/data/project-dyd</u>, with a symbolic link pointing to it in the home directory. For any hosting-related issues or clarifications, it is recommended to reach out to Chris, who is well-versed in the server configuration.

# 4. Deployment and Maintenance

The project maintenance process is streamlined through Git, with the development branch designated for hosting updates and version control. Any changes or new features should be pushed to this branch and then redeployed as needed. For deployment, the build folder (generated using npm build) must be uploaded to the /home/project-dyd/design-your-degree directory on the server via FileZilla, which updates the hosted site at https://design-your-degree.research.cs.dal.ca/.

Although we initially encountered challenges while hosting, Chris Maxwell from the FCS Help Desk provided significant guidance on the server structure and the deployment process.

The actual directory being served is <u>/local/data/project-dyd</u>, with a symbolic link pointing to it in the home directory. For any hosting-related issues or clarifications, it is recommended to reach out to Chris, who is well-versed in the server configuration.

Access to the software repository is managed through the FCS GitLab; for access permissions, please contact <a href="mailto:fcsproj@dal.ca">fcsproj@dal.ca</a>

#### **Deployment Steps:**

Log in to the server via FileZilla using

• Username: project-dyd

• Password: Oily6Goose6Water are correctly set in your `.env` file.

Navigate to: /home/project-dyd/design-your-degree/ Upload the newly generated build folder (from npm run build) to overwrite and update the live site.

# 5. Client Feedback and Next Steps

#### **Client Feedback:**

Stephanie provided helpful and actionable feedback. She requested:

- Cleaner, uniform box sizing for better layout and printability.
- Displaying all semesters per year for clear course planning, including placeholders for non-study terms.
- Visual and pop-up alerts for unmet prerequisites and duplicate course entries.
- Consistent Dalhousie branding, preferably using the FCS logo.
- Print functionality and clickable links to course descriptions for easier navigation and planning.

# **Next Steps:**

- Develop separate planners for BACS and BCS students.
- Implement login functionality to store user info in the FCS database).
- Complete backend deployment, including WSGI setup and populating MariaDB.
- Integrate links to Dalhousie course descriptions.
- Improve how co-op and non-study terms are represented in the planner.

# 6. Lessons Learned & Recommendations

#### **Lessons Learned:**

- 1. **Keeping Code Organized Made Things Easier:** Splitting the code into different files for each feature (like CourseCard.jsx and DegreePlanner.jsx, etc.) helped a lot. It allowed team members to work separately without causing confusion and made debugging quicker.
- 2. **Git Workflow Was Mostly Good, But Can Be Better:** Using branches and pushing regularly helped the team work together smoothly. But in the future, it's better to follow proper naming, always branch from Development, and keep the branch up to date to avoid any mess or merge conflicts.

- 3. Communication Within the Team Is Very Important: We learned that regular checkins and being open about blockers or confusion helped us avoid delays. Whenever someone got stuck, quick discussions saved a lot of time instead of waiting or guessing.
  - Furthermore, proper communication fosters a more productive environment, by allowing everyone express themselves openly which in turn boosts morale and motivation.
- 4. **Dealing With Merge Conflicts Takes Practice:** We ran into a few Git merge conflicts, and learning how to resolve them calmly (without panicking or breaking the project) was a good skill we picked up during this project.
- 5. **Scoping Realistically Is Key:** Some features had to be scaled back due to time constraints. We learned to distinguish between key features and supplementary improvements, which helped us prioritize our efforts and deliver a functional product within the deadline.
- 6. Client Feedback Drives the Most Valuable Changes: We learned that listening and properly implementing the client's feedback is essential to deliver a useful product, and avoid unnecessary rework, and stay efficient during development.

#### **Recommendations for Future Teams:**

1. Add Secure Login and Hashing:

Security can be improved by adding hashed password storage and proper login systems like Firebase Auth or OAuth would work well.

2. Make the UI Mobile-Friendly and Accessible:

Some parts of the UI can be improved for smaller screens and accessibility. Making the tool work better on phones and tablets would be useful.

3. Add filtering by department:

Implement sorting/filtering by department and/or difficulty for better user experience.

#### 7. Team Contributions

This project was made possible through the collective effort of each team member, with specific contributions helping shape the outcome. The User Interface and overall website design were primarily led and implemented by **Dhruvi**. For any inquiries or clarifications related to UI components or design elements, please feel free to reach out to her directly at: <a href="mailto:dh777915@dal.ca">dh777915@dal.ca</a>. On the backend side, including API setup, integration, and database connectivity, **Harshil Vipul** 

**Lotwala** played a key role. He can be contacted for any technical questions related to the backend architecture and database management using his dal email id: <a href="https://hrefs.ncbi.nlm.nih.good.ncbi.nlm