

1 Prerequisites

- Java 17
- Gradle 8.6
- Network should be stable for downloading dependencies
- VSCode

2 General Setup

2.1 Cloning the Project from GitLab

```
git clone -b git@gitlab.oit.duke.edu:kits/project-000-000-01-test/project-team-4.git
```

2.2 Building Projects

Change directory and build the whole project. If your gradle version is too old, you can change gradle-wrapper.properties and use gradle wrapper to update it.

```
./gradlew build
```

3 Frontend

3.1 Configuration

We use React to develop frontend of website. The required version of node should be higher than 18.17.0 and npm version should be higher than 10.19.0

3.2 How to Set Up the Environment

```
sudo apt-get remove nodejs
sudo apt-get purge nodejs
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.1/install.sh | bash
# Reload your shell
exec $SHELL
# Install a newer Node.js version
nvm install 18
nvm use 18
npm install -g npm@latest
node -v
npm -v
npm install react-scripts --save
```

4 Backend

4.1 Configuration

We use spring boot with version of 3.24.6. You can just find the TrackerServer-Application.java and simply run. Everything in the application.properties will be downloaded automatically.

4.2 Running the Application

Our frontend and backend service can run on different machines, so in order to enable cors, you should change WebConfig.java to your own server.

```
.allowedOrigins("http://vcm-38068.vm.duke.edu:3000")
```

5 Desktop JavaFX

5.1 Configuration

Configure the Course Administration module by ensuring compatibility of all software versions. The 'build.gradle' file contains the critical version information for the setup.

5.2 Version Specifications

- **Java Version:** The module is compatible with Java 17, supporting modern Java features and APIs.
- **Gradle Plugin Versions:**
 - **JavaFX Plugin:** Version 0.0.10 - Facilitates managing JavaFX dependencies and streamlines the build process for JavaFX-based applications.

5.3 Dependencies

The Course Administration module relies on various external libraries for essential functionalities:

- **JUnit:**
 - **JUnit Jupiter API:** Version 5.7.0 - Used for crafting test cases in the JUnit 5 format.
 - **JUnit Platform Launcher:** Version 1.6.2 - Enables the JUnit platform to run in diverse environments.
- **Mockito:** Version 4.0.0 - Offers tools for mocking in unit tests, vital for isolating component testing.

5.4 Running the Application

To launch the Course Administration application, execute the following command:

```
./gradlew run-fx
```