

# Week 3 Milestone 3 Report

Group Name: Team Q

Members: Lita Hernandez-Gonzalez, Purani Daivik Manishkumar, Jannelly Hernandez  
Umanzor, Andrew Guinto

Link to Github repository:

<https://github.com/sfsu-csc-667-fall-2024-roberts/term-project-team-q>

Link to Github project board:

<https://github.com/orgs/sfsu-csc-667-fall-2024-roberts/projects/11/views/1>

## **A brief description of what we accomplished in the term project this week:**

This week, our term project work concentrated on establishing the front-end environment for the application, which included configuring essential tools to facilitate development and module management. We also set up routes for the pages that the game has and added templates to those routes.

### **Accomplishments:**

1. Frontend Configuration with Webpack: We successfully configured Webpack to bundle TypeScript files, thereby simplifying the management of dependencies and module loading on the front end. This process included the creation of `webpack.config.ts`, which outlines the compilation of our TypeScript files and specifies the location for the final build outputs.
2. Build Automation: We incorporated build scripts into `package.json` to automate the build process. This includes a development script, `build:dev`, which monitors for changes and triggers rebuilds as necessary, thus enhancing the efficiency of front-end development.
3. Concurrent Process Management: We integrated `concurrently` to facilitate the simultaneous execution of both server (`nodemon`) and front-end (`webpack`) processes, optimizing the development workflow.
4. Route Management: We set up routes for all the pages that the game intends to have.

### **Milestones:**

1. Configure Webpack for TypeScript Bundling: Develop a `webpack.config.ts` file to consolidate TypeScript files, resulting in a single bundled output in the public directory.  
Automate Builds: Establish `build` and `build:dev` scripts within `package.json` to manage production and development builds, respectively.
2. Set Up Liveload for Development: Integrate `livereload` and `connect-livereload` to enable automatic page reloads in response to code changes.  
Implement Concurrent Execution: Modify `start:dev` in `package.json` to run `nodemon` and `webpack` concurrently, thereby minimizing the complexity of managing separate processes.
3. Set up all routes: Add routes and add template pages for all pages.