Avoidance: The Game

User Documentation

**About This Project**

Avoidance: The Game is a decision making game in which the player tries to go about their daily life while avoiding contact with other people during a global pandemic while maintaining their physical and social health.

Main Features

* Playable over the internet with no installation required.
* Multiple levels to play through.
* Play as a Nurse, Teacher, or Librarian each with different starting health stats.
* Players can have their name saved to a leaderboard.
* Dual health and social status bars with feedback when health is lost or gained.

**Installation And Usage**

If you just want to play the game you may do so at <https://avoidance-the-game.herokuapp.com/>.

To install and run locally complete the following steps:

1. **Install the game files and node.js**
   * Install [node.js](https://nodejs.org/) on your machine.
   * Download and unzip a release from the [Avoidance Github](https://github.com/Ada-Kru/Avoidance-The-Game/).
   * Open a command line in the same directory where you unzipped the Avoidance files.
   * Run 'npm install' and the appropriate node modules will be installed.
2. **Install the database server**
   * Install PostgreSQL on your machine from <https://www.postgresql.org/>.
     + You will be asked to choose a password during installation. Remember this as it will be needed later.
   * Open a command line in the 'bin' directory within the directory where you chose to install PostgreSQL.
   * Run 'psql' and choose the user 'postgres' and use the password you chose during installation.
     + You should be inside the PostgreSQL command line environment. Run 'CREATE DATABASE Avoidance;' without quotes but with the semicolon.
     + Run '\connect Avoidance' to connect to the database.
     + Run the following command (you should just be able to copy and paste it into the environment.)

CREATE TABLE scores(

id SERIAL PRIMARY KEY,

name VARCHAR(32) NOT NULL,

character\_type INT NOT NULL,

total\_score INT NOT NULL DEFAULT -10000,

health\_score INT NOT NULL DEFAULT -10000,

social\_score INT NOT NULL DEFAULT -10000,

created TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

* + Run '\q' to exit the psql environment.
* In order for the node server to connect to the database, environment variables must be set on your system. The following instructions are for running on Windows. Setting environment variables is different on different operating systems so you may need to look up a guide on how to do this if you are using a different operating system.
* On Windows there is a 'pg\_env.bat' file in the installation directory for PostgreSQL. You can copy this to another directory, modify it as needed, and run it before you start the node server. The following variables need to be set (the password and port might be different depending on your choices during installation):

PGHOST=localhost

PGDATABASE=Avoidance

PGUSER=postgres

PGPASSWORD=your\_password\_here

PGPORT=5432

DATABASE\_URL=postgres://localhost/Avoidance

1. **Run the server**

* You must run the node server from the same command terminal that you set the environment variables in. So if you set the environment variables in the batch file on windows you would run ‘pg\_env.bat’ first and then run the command to start the node server.
* To start the server run 'node index.js' in the directory where your Avoidance files are installed.
  + To stop the server press ctrl-c in the terminal where the server is running.

1. **Play the game**

* Navigate to <http://localhost:5000/> in a modern web browser to play the game.

**Quality Attributes**

This project adheres to the following quality standards:

**Playability** - The game should be enjoyable to play.

**Reliability** - All the features and functions should work 100% of the time.

**Usability** - The game should be easy to use.

**Architecture And Design**

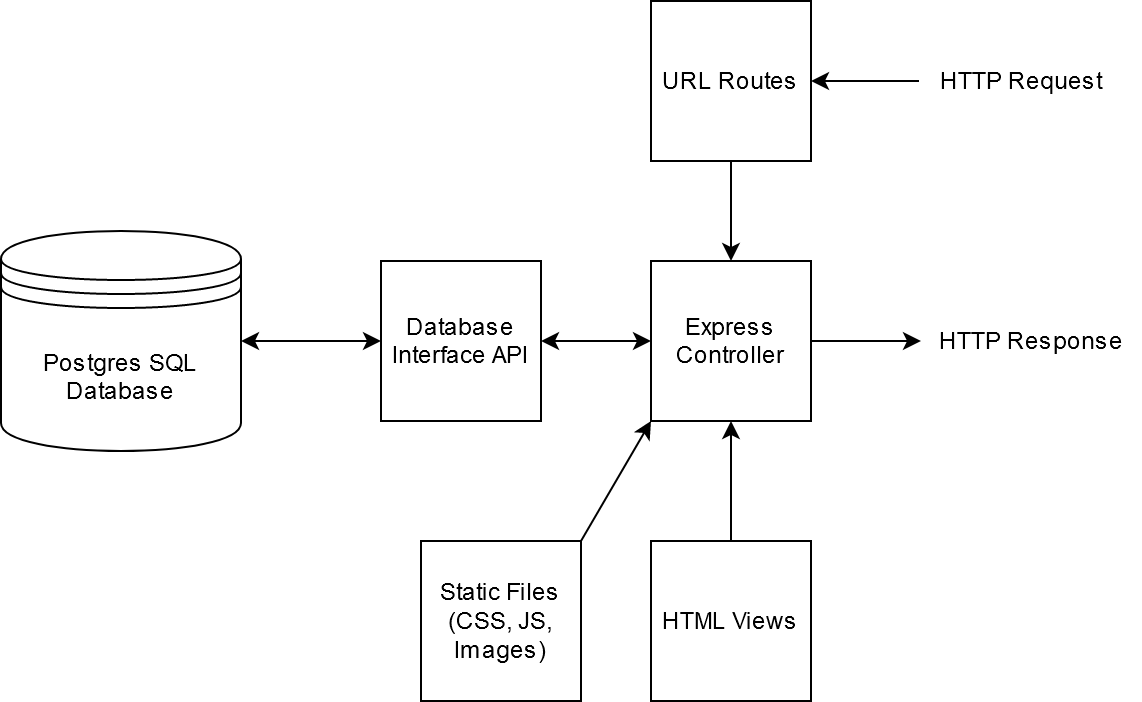
The program is a web application coded in JavaScript. The server is run with node.js and the back end is coded using the Express framework. Node allowed us to use modern JavaScript features such as asynchronous programming without worrying about compatibility issues with web browsers. The Express framework allowed us to keep our source files organized in a way that reduced the temptation to do too much in one file. Express also provided us with a fully featured server that is widely used enough that there was plenty of documentation when we ran into problems and needed to find a solution.

The front end scripts are written in vanilla JavaScript and jQuery. Using jQuery helped simplify some of the more tedious parts of changing the HTML code dynamically when navigating the levels in the game. The page layouts are created using standard HTML and CSS. We tried not to rely too much on third party libraries so as to reduce the time it would take to complete the project. This also helped to ensure that the game ran quickly and we had a large amount of information available when we needed to troubleshoot problems. Data for the levels is stored in its own JavaScript file and loaded as a variable into the main game play file. The main game play file loads the appropriate parts of the data variable according to which level the player is currently on. The front end code takes advantage of the modern fetch API to send and retrieve data from the database. The fetch API uses promises and we were able to use asynchronous programming to avoid using callbacks. This helped us to keep the front end code concise and orderly.

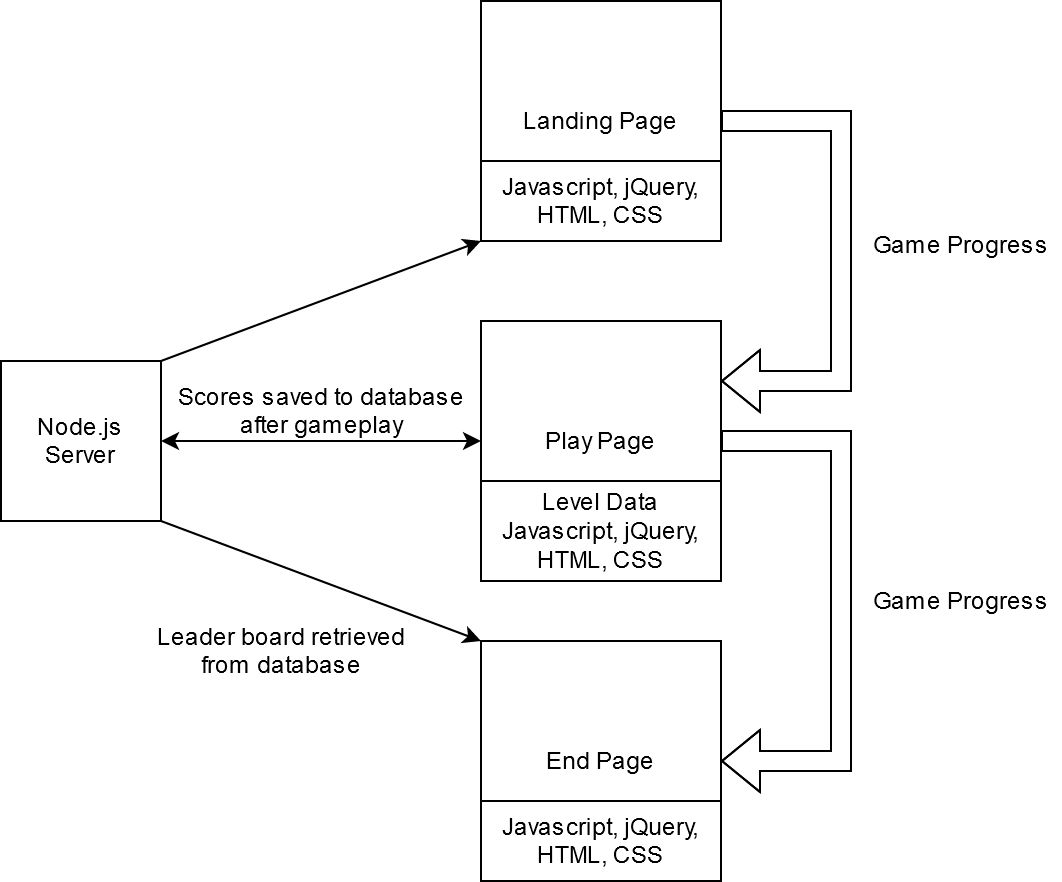
For the database we used PostgreSQL. We chose this because it was the easiest database to set up on Heroku and it is widely used and well documented. To interface with the back end we used the ‘pg’ module for node.js. This module supports asynchronous programming which helped keep the code clean due to providing us with the ability to avoid using callback functions. The database’s settings are saved in environment variables. This allows developers to use different database settings on different systems without needing to change the game’s source code. The front end code is able to send and retrieve data to and from the database through HTTP requests with a routing API created in Express.

We deploy the application using Heroku. Heroku provides servers and hosting for free which was ideal for this project. The major downside is that for the free tier there is a delay when opening the application if it has not been used in a while. Heroku also provides command line tools that allow for remote configuration which proved useful when configuring the database. We are able to quickly update the code on the Heroku server due to it having the ability to load code from our GitHub repo remotely.

**Back end Architecture Diagram**



**Front end Architecture Diagram**

****

**Design Patterns**

The back end database API module uses the observer design pattern to monitor users saving their scores in order to alert them if they got the high score.

The back end database API module also uses the adapter design pattern in order to adapt incoming HTTP requests into PostgreSQL database requests.

**Final State Of The Project**

Known Bugs

* There may be a few rendering problems when playing the game through Safari.

Product Backlog

* Add more playable levels.
* Add the ability for levels to have branching paths.

**File Structure**

**Main directory** – Contains the index.js file used to run the server and other data files needed by node and the express server.

**db** – Contains source files related to the database interface.

**public** – Directories and files accessible through HTTP requests. All the script, CSS, and image files are stored here.

**routes** – Source files for sending and receiving data to and from the database through HTTP requests.

**util** – Text documents describing how to run and use the application and scripts used for maintenance of the application.

**views** – Contains the HTML files for the application.

**How To Contribute In The Future**

Current team members and new developers are welcome to contribute to the project in the future. The project will remain hosted at its current location: <https://github.com/Ada-Kru/Avoidance-The-Game/>. To contribute pull requests may be submitted to github and all current team members will retain the ability to accept or reject them after review. Submissions should be well written and meet the project’s quality attributes described above to be accepted.