

Instructions Set Up Ballhog (trade table)

Windows Setup:

1. Install Node.js:

- Download Node.js from <https://nodejs.org/>
- Choose the LTS version and run the installer
- Follow the prompts to install Node.js and npm

2. Install MySQL:

- Download MySQL Community Server from <https://dev.mysql.com/downloads/mysql/>
- Run the installer and follow the prompts
- Set a root password during installation
- Add MySQL to the system PATH if prompted

3. Set up the project:

- Unzip the project folder from the Downloads folder
- Open Command Prompt and navigate to the unzipped project folder
- Run these commands:

```
npm init -y
npm install express mysql2 node-cron cors @google/generative-ai googleapis
```

4. Set up the MySQL database:

- Open MySQL Command Line Client

- Enter your root password
- Run these SQL commands:

```
CREATE DATABASE ballhog_googlespreadsheet;  
USE ballhog_googlespreadsheet;  
CREATE TABLE ballhog_projections (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  `rank` INT,  
  player_name VARCHAR(255),  
  position VARCHAR(50),  
  team VARCHAR(50),  
  games_played INT,  
  minutes_per_game FLOAT,  
  field_goals_made FLOAT,  
  field_goals_attempted FLOAT,  
  field_goal_percentage FLOAT,  
  free_throws_made FLOAT,  
  free_throws_attempted FLOAT,  
  free_throw_percentage FLOAT,  
  three_pointers_made FLOAT,  
  points_per_game FLOAT,  
  rebounds_per_game FLOAT,  
  assists_per_game FLOAT,  
  steals_per_game FLOAT,  
  blocks_per_game FLOAT,  
  turnovers_per_game FLOAT,  
  z_scores FLOAT  
);
```

5. **Configure database connection in `server.js` : (for me the default is root and password: passsword)**

```
const pool = mysql.createPool({  
  host: 'localhost',  
  user: 'root',
```

```
password: 'your_mysql_root_password',  
database: 'ballhog_googlespreadsheet',  
// ... other options ...  
});
```

6. Set up Google Sheets API:

- Ensure `Ballhog IAM Admin.json` is in the same directory as `server.js` (should already be there cause its zipped)

7. Run the server:

- In Command Prompt, navigate to the project folder
- Run: `node server.js`

8. Access the application:

- Open a web browser and go to <http://localhost:3000>

The instructions and links will also be given in the terminal window after the command is run. Note the pulling from google sheet happens every hour

1. Set up environment variables: (the gemini feature doesn't work yet, so can be skipped**)

- Create a `.env` file in the project root
- Add: `GEMINI_API_KEY=your_gemini_api_key_here`

2. Run the server:

- In Command Prompt, navigate to the project folder
- Run: `node server.js`

3. Access the application:

- Open a web browser and go to <http://localhost:3000>

macOS Setup:

1. Install Node.js:

- Download Node.js from <https://nodejs.org/>
- Choose the LTS version and run the installer
- Follow the prompts to install Node.js and npm

ORRRR

For macOS:

1. Install Homebrew (if not already installed):

```
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

2. Install Node.js and npm:

```
brew install node
```

3. Verify the installation:

```
node --version  
npm --version
```

4. Install MySQL:

- Download MySQL Community Server from <https://dev.mysql.com/downloads/mysql/>
- Run the installer and follow the prompts
- Note the temporary root password provided during installation

5. Set up the project:

- Unzip the project folder from the Downloads folder
- Open Terminal and navigate to the unzipped project folder
 1. If your project folder is zipped, you'll need to unzip it first. Assuming the zip file is named "fantasysite.zip", you can use these commands:
 2. Navigate to the Downloads folder:

```
Copy
cd ~/Downloads
```

3. Unzip the file:

```
Copy
unzip fantasysite.zip
```

4. Then navigate into the unzipped folder:

```
Copy
cd fantasysite
```

5. To verify that you're in the correct directory, you can use the `pwd` command (print working directory):

This should output something like:

```
/Users/yourusername/Downloads/fantasysite
```

```
pwd
```

6. To see the contents of the folder, you can use the `ls` command:

This will list all files and folders in your current directory.

Remember, macOS terminal commands are case-sensitive, so make sure you type the folder name exactly as it appears.

- Run these commands:

```
npm init -y
npm install express mysql2 node-cron cors @google/generative-ai googleapis
```

6. Set up the MySQL database:

- Open Terminal and run: `mysql -u root -p`
- Enter the temporary root password
- Set a new root password: `ALTER USER 'root'@'localhost' IDENTIFIED BY 'your_new_password';`
- Run these SQL commands:

```
CREATE DATABASE ballhog_googlespreadsheet;
USE ballhog_googlespreadsheet;
CREATE TABLE ballhog_projections (
  id INT AUTO_INCREMENT PRIMARY KEY,
  `rank` INT,
  player_name VARCHAR(255),
  position VARCHAR(50),
  team VARCHAR(50),
  games_played INT,
  minutes_per_game FLOAT,
  field_goals_made FLOAT,
  field_goals_attempted FLOAT,
  field_goal_percentage FLOAT,
  free_throws_made FLOAT,
  free_throws_attempted FLOAT,
  free_throw_percentage FLOAT,
  three_pointers_made FLOAT,
  points_per_game FLOAT,
  rebounds_per_game FLOAT,
  assists_per_game FLOAT,
  steals_per_game FLOAT,
  blocks_per_game FLOAT,
  turnovers_per_game FLOAT,
  z_scores FLOAT
);
```

7. Configure database connection in `server.js` :

```
const pool = mysql.createPool({
  host: 'localhost',
  user: 'root',
  password: 'your_new_password',
  database: 'ballhog_googlespreadsheet',
  // ... other options ...
});
```

8. Set up Google Sheets API:

- Ensure `Ballhog IAM Admin.json` is in the same directory as `server.js`

9. Set up environment variables:

- In Terminal, run: `nano .env`
- Add: `GEMINI_API_KEY=your_gemini_api_key_here`
- Save and exit (Ctrl+X, Y, Enter)

10. Run the server:

- In Terminal, navigate to the project folder
- Run: `node server.js`

11. Access the application:

- Open a web browser and go to <http://localhost:3000>

For both setups, ensure all necessary files (HTML, CSS, JavaScript, and the Google service account JSON file) are included in the project folder. The Cron job for pulling Google Sheets data is already set up in `server.js` and will run automatically every hour when the server is running.