Wordio Game

Introduction

Thank you for purchasing code. It really means a lot and It's our pleasure to serve top-notch service to you. If you have trouble with the code and documentation please contact to our Support Team here

Installation of NodeJS and NPM

Nodejs *is an open-source, cross-platform runtime environment for developing server-side web applications. Nodejs also has an event-driven architecture capable of asynchronous I/O. Nodejs uses an event-driven, non-blocking I/O model that makes it lightweight and efficient.*

npm *(abbr. node package manager) is a standard package manager that is automatically installed together with Node.js. It is used to download packages from the npm cloud server, or to upload packages to these servers.*

Installation of **Nodejs** and **npm** is straightforward using the installer package available at Nodejs official web site.

- Download the installer from Nodejs.
- Run the installer.
- Follow the installer steps, agree the license agreement and click the next button.
- Restart your system/machine.

Now, test Nodejs by printing its version using the following command in Command Prompt:

```
node -v
```

and test npm by printing its version using command

```
npm -v
```

You are done with installation.

Installation VS Code

- Visual Studio Code is a lightweight but powerful source code editor.
- For Download VS Code click here
- Open VS Code then click `File > Open Folder..` and choose folder with project

Install dependencies

Packages in Node.js

In short, a package in Node.js is one or several JavaScript files representing some kind of library or tool.

Once you have dragged your project into your code editor, you can open up your terminal (in VSCode, go to `View > Terminal`).

Run this command to install dependencies :

```
npm i
```

> *What does **npm i** do? npm will download the packages to the node_modules project folder according to the configuration in the package. json, updated the package version wherever possible (and, in turn, updated package-lock. json)*

Project Structure

Once our project files have been created and our dependencies have been installed, our project structure should look like this:

wordio-game

- node_modules
- package.json
- .gitignore
- public
- src
- README.pdf
- README.md

What are each of these files and folders for?

- **node_modules** is a folder that includes all of the dependency-related code that Create React App has installed. You will never need to go into this folder.
- **package.json** that manages our app dependencies and what is included in our node_modules folder for our project, plus the scripts we need to run our app.
- **.gitignore** is a file that is used to exclude files and folders from being tracked by Git. We don't want to include large folders such as the node_modules folder
- **public** is a folder that we can use to store our static assets, such as images, svgs, and fonts for our React app.
- **src** is a folder that contains our source code. It is where all of our React-related code will live and is what we will primarily work in to build our app.
- **README.pdf**, **README.md** these are the documentation files.

Run the React Application

Now you are ready to run your application!

Open up your terminal (in VSCode, go to `View > Terminal`).

Run this command to run the React application wordio-game:

```
npm start
```

A new browser window will pop up with your React App! If not, open your browser and type localhost:3000 in the address bar.

Modify the React Application

So far so good, but how do I change the content?

Look in the wordio-game directory, and you will find a src folder. Inside the src folder there is a file called App.jsx, open it, try changing the content and save the file.

> *Notice that the changes are visible immediately after you save the file, you do not have to reload the browser!*

How to Build the App

Once we are happy with our app and are ready to publish it, we can build it with the following command:

```
npm run build
```

This command will create an optimized production build for our project and will output what files it has generated and how large each file is: The output is coming from the build tool Webpack.

It helps to give us an idea of the size of our app files because the size of our .js files in particular can make a large impact on our app's performance.

Each chunk includes a unique string or hash, which will change on every build to make sure any new deployment is not saved (cached) by the browser.

If we did not have this cache-busting hash for each of our files, we likely couldn't see any changes we made to our app.

Finally, we can run our built React project locally with the help of the npm package serve.

This is helpful to detect any errors we might have with the final version of our project before pushing live to the web.

Like create-react-app, we can use npx to run serve without installing it globally on our computer.

```
npx serve -s build
```

Using serve, our app will start up on a different development port instead of 3000. In this case, localhost:5000.

And with that, we have a completed React application ready to publish live to the web on any deployment service, such as Vercel, Netlify, or Heroku.

How to Deploy App with Vercel

Vercel is a platform for deploying the fastest React sites. You can deploy your site with zero configuration to the best frontend infrastructure.

- Develop: Build React sites that connect to your favorite APIs, databases, and content management systems.
- Preview: Integrate with any GitHub, GitLab, or Bitbucket repository for instant continuous deployment.
- Ship: Deploy your site to every edge node worldwide for the fastest React sites. Static files, Serverless and Edge Functions, and more.
- Monitor: Measure Core Web Vitals from actual devices your visitors are using with Vercel Analytics for Next.js or Gatsby.

Built-in CI/CD for React sites

Vercel has integrations for GitHub, GitLab, and Bitbucket to enable CI/CD for your React site with zero configuration. Then, you can run automated tests for performance and reliability on every push. Pull and merge requests are deployed instantly to a unique URL, accessible to your entire team.

Vercel CLI

- Install the Vercel CLI and run vercel to deploy.
- Vercel will detect that you are using React and will enable the correct settings for your deployment.
- Your application is deployed! (e.g. create-react-template.vercel.app)

Vercel for Git

- Push your code to your git repository (GitHub, GitLab, BitBucket).
- Import your React project into Vercel.
- Vercel will detect that you are using React and will enable the correct settings for your deployment.
- Your application is deployed! (e.g. create-react-template.vercel.app)

Add your custom domain

After deploying, your new React site will automatically be assigned a `.vercel.app` suffixed domain. You can then add a Custom Domain on your choice, either from a third-party or purchased through Vercel.

Support

Its our pleasure to serve our service and support. please contact our support team. Head of Customer Support: Telegram Typically we reply our customers for all the questions and queries within 24 hours of time via comments, support forum or emails.

Thank you very much.