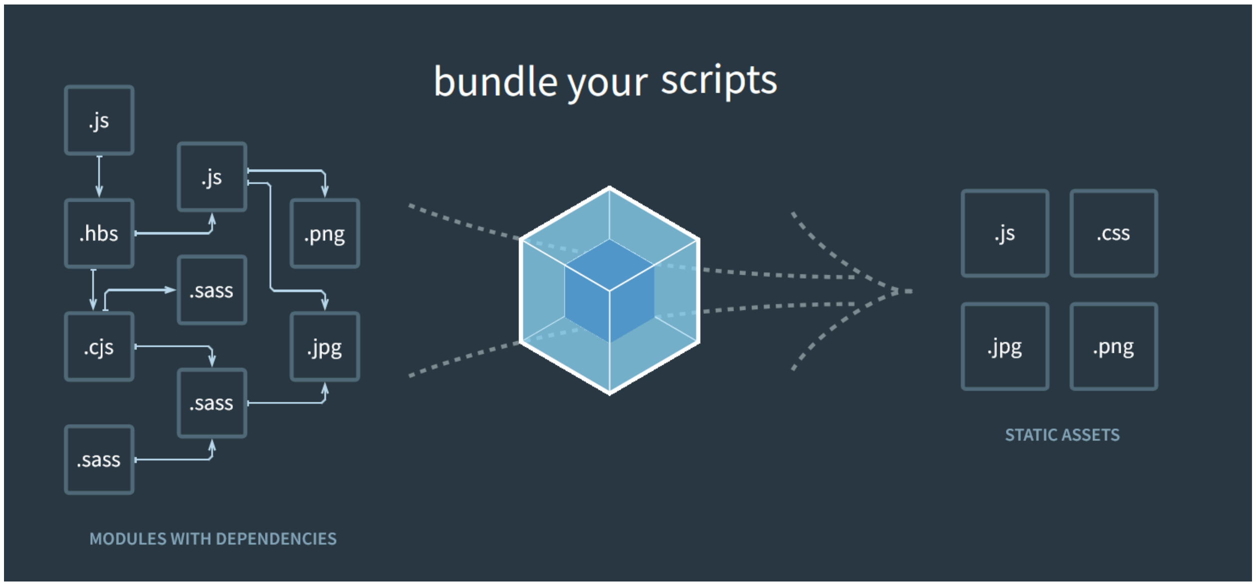
React Assignment 2

1) What is NPM?

NPM (Node Package Manager) is an package manager for Node Js, meaning its like an app store in mobiles or computers which has many apps for them
And just like that, NPM has many libraries/packages for the Node Js framework
And each library/package is for a different purpose.
It contains many prebuilt libraries and packages to install and use.
Also one developer can create their own library/package and can upload to NPM so others can use them.
You can install, update, and uninstall packages using NPM.

2) What is Parcel/Webpack ? Why do we need them?

These are called module bundlers, and used to bundle all the JS files, CSS etc (Source Code) into one bundle/large file (Production Code) and then that file can be used in the browser and load the JS app/JS Bundle
The JS Bundle contains the source code, any third party dependencies that we imported etc
Bundler like webpack, creates a dependency graph to track all the dependencies and how to put all the dependencies together. We need to tell the entry point and the bundler will go through the imports etc and try to pack everything into one single file.



3) What is parcel cache ?

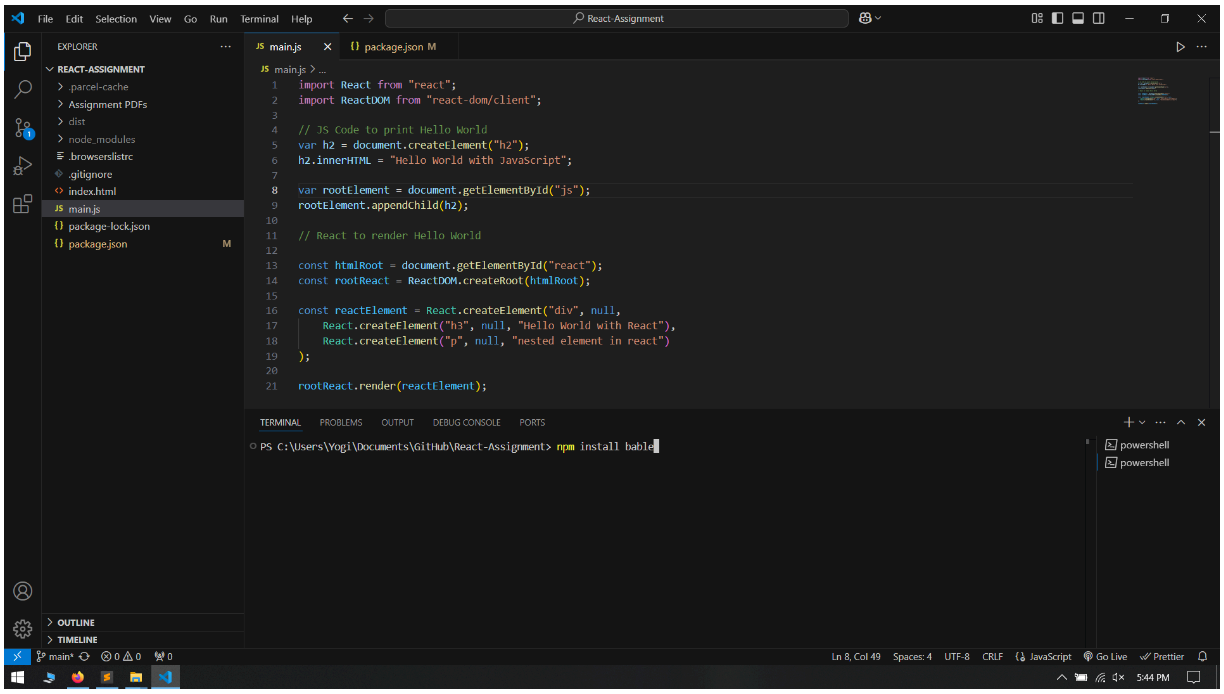
The parcel cache stores all info about the project when we run the first time, thats why it will take a longer time the first time we run the project, downloading/storing all the important info to the cache folder/file so that when we run the project 2nd time, it will use the cache folder to run the project instead of re-parsing/scanning/analyzing again everything from start.
Making it fast and easy for developers.

4) What is NPX?

NPX stands for **Node Package Execute**. It's a command-line interface (CLI) tool that comes bundled with **npm** (starting from version 5.2.0 and above). When you install npm, NPX is installed automatically.
NPX allows you to run JavaScript packages directly from the npm registry without needing to install them globally or locally on your system. Think of it as a "package runner" that executes npm packages straight from the registry.
While **npm** is primarily used for installing packages (either globally or locally), **npx** is focused on executing them without adding them permanently to your system.

5) Difference between dependencies and devDependencies?

Dependencies:
Its a library that a project needs to work properly
We need these libraries when we run the code
When you deploy your project, these dependencies are included in the final bundle.
We always need these to run the application
Installing the dependencies:



```js
import React from "react";
import ReactDOM from "react-dom/client";

// JS Code to print Hello World
var h2 = document.createElement("h2");
h2.innerHTML = "Hello World with JavaScript";

var rootElement = document.getElementById("js");
rootElement.appendChild(h2);

// React to render Hello World

const htmlRoot = document.getElementById("react");
const rootReact = ReactDOM.createRoot(htmlRoot);

const reactElement = React.createElement("div", null,
    React.createElement("h3", null, "Hello World with React"),
    React.createElement("p", null, "nested element in react")
);

rootReact.render(reactElement);
```

```
PS C:\Users\Yogi\Documents\GitHub\React-Assignment> npm install bable
```

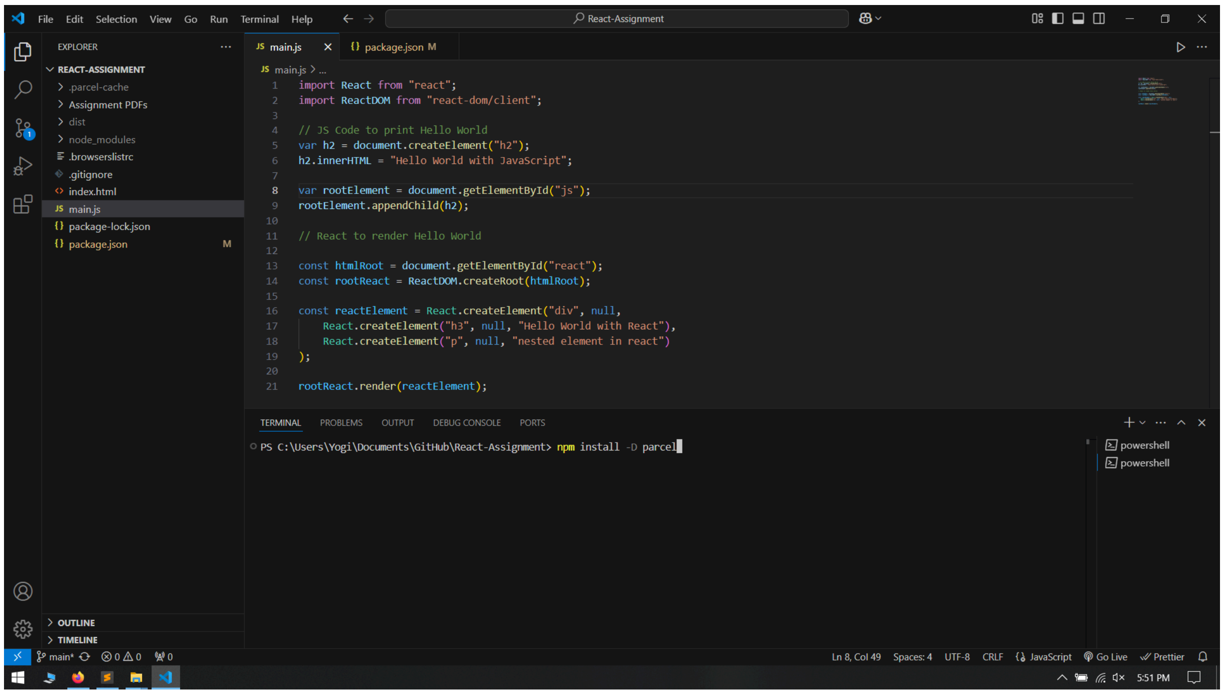
devDependencies:

These are also libraries that are used by the developers.

But these are only required when like developing the application and doesnt need after the production of the application.

So basically these dependencies are used in the development to test or develop the application and to make development a bit easy, and after that the fully developed application doesnt need these to run, so normally when the final bundle is produced the dev dependencies are removed/cleaned

Installing the devDependencies:



```javascript
import React from "react";
import ReactDOM from "react-dom/client";

// JS Code to print Hello World
var h2 = document.createElement("h2");
h2.innerHTML = "Hello World with JavaScript";

var rootElement = document.getElementById("js");
rootElement.appendChild(h2);

// React to render Hello World

const htmlRoot = document.getElementById("react");
const rootReact = ReactDOM.createRoot(htmlRoot);

const reactElement = React.createElement("div", null,
    React.createElement("h3", null, "Hello World with React"),
    React.createElement("p", null, "nested element in react")
);

rootReact.render(reactElement);
```

```
PS C:\Users\Yogi\Documents\GitHub\React-Assignment> npm install -D parcel
```

6) What is Tree Shaking ?

It is way to remove unnecessary code / dead code from the project

This makes the project smaller and easy to run

Uses ES modules to see what code is used and what is not is used

Some bundlers trace/track our code by making a tree structure

And then the bundle shakes this tree, removing any code etc that are not needed.

7) What is Hot Module Replacement?

Hot Module Replacement (HMR) is a feature in module bundlers/builders like webpack, parcel and vite, etc

When we make changes to our source code or module the build tool detects the change and instead of rebuilding the full entire app, the HRM will only updates the specific code/module that changed and when the browser receives the module, it will replace the specific module / part of the site, instead of reloading the whole page.

8) List down your favourite 5 superpowers of Parcel and describe any 3 of them in your own words?

Dev server:

In parcel there is an inbuilt server which is used to run the app and makes it easy because we just run by typing NPX parcel index.html

And also it supports the HTTPS protocol which makes the app more secure.

HMR:

Hot module replacement is one of my favourite feature because the way it works like detecting the changes and updating the specific code/module and when the browser gets the module or code part, it will replace the specific module of the site instead of reloading the whole page is interesting

Caching:
As the parcel runs the bundled app etc, and the data will be stored in the cache folder and after the first run of the app, parcel uses the cache folder instead of re-parsing the whole source code.

Tree Shaking:

Tree shaking is a way to remove unnecessary code / dead code from the project

Uses ES modules to see what code is used and what is not is used

Some bundlers track our code by making a tree structure

And then the bundle shakes this tree, removing any code etc that are not needed.

Plugins:

Plugins allows us to expand the functionality of the parcel by adding plugins to it and projects

Plugins can be installed from npm and then added in your Parcel project.

We can use a `.parcelrc` file to specify plugin configurations globally or use within the code

8) What is .gitignore ? what should we add and not add to it?

Its a file which is used to tell git or git clients about what files we want to add to our repo and whats not and we should add the file to the root of the project/folder

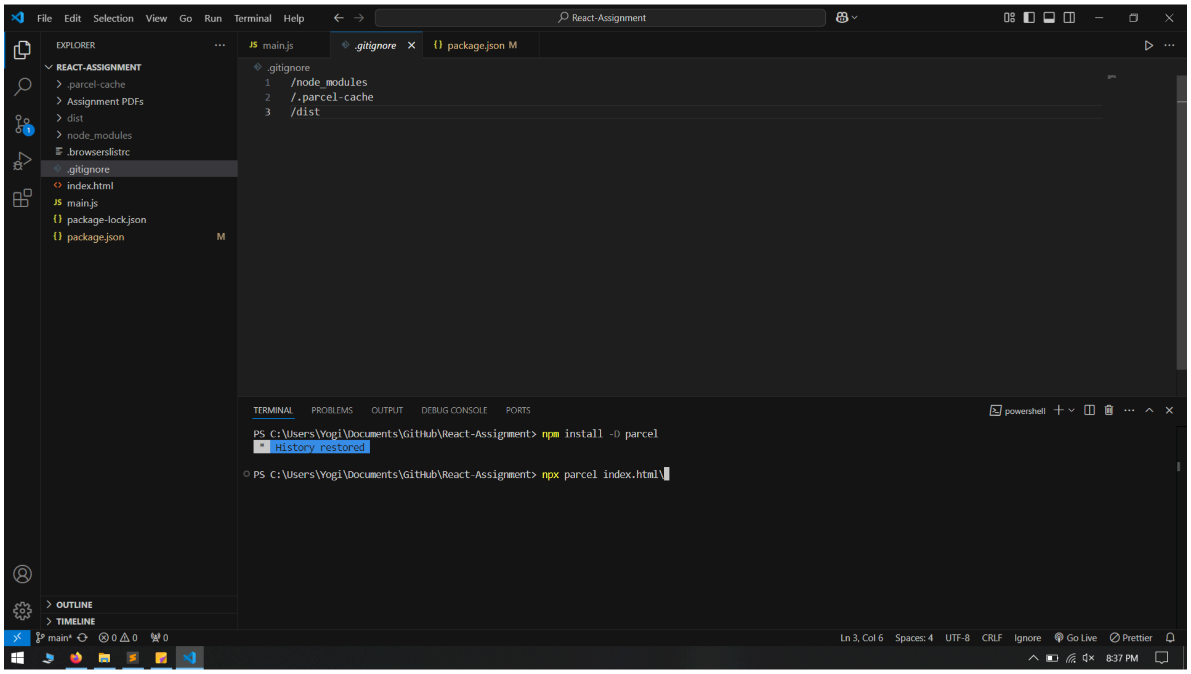
We should add the folders and files which are larger in size or more number of files that we dont need and also sometimes the dev dependencies files that we don't want to get uploaded to the git repo.

We can use (/) at the starting or at the end of the folder name to ignore the folders which are present in the root folder

We can also use (*) and then the file extension to ignore the same type of files like (.class) files etc

And also we can write the whole file name to ignore that specific file from uploading git

Example:



```
/node_modules
/.parcel-cache
/dist
```

```
PS C:\Users\Yogi\Documents\GitHub\React-Assignment> npm install -D parcel
 *  History restored

PS C:\Users\Yogi\Documents\GitHub\React-Assignment> npx parcel index.html\
```

9) Difference between package.json and package-lock.json?

package.json:

Its the main config file in a node project, which tells about the project's metadata(data about the data) dependencies etc

It tells the packages and their versions which are being used in the project.

It also tells about what dependencies are for production and what devDependencies are foe development environment.

Package-lock.json:

This file will be generated automatically when we install or update dependencies using npm

Tells/records about the exact version of each installed package/dependencies which allows you to re-install them.

This file makes sure that everyone that works on the project uses the same dependencies to build consistent project together.

10) Why should i not modify the package-lock.json file?

If we modify the package lock file, it can cause inconsistent versions of dependencies and can cause issues in our project.

it can cause Inconsistent builds, security risks etc

So basically as the Parcel will take care of it we shouldnt modify it without knowing how and what to change, we can use package.json instead if we want to do something like adding or removing dependencies.

11) What is node_modules ? is it a good idea to push it on to git?

Its a folder which downloads and stores all the packages and the dependencies that are required or downloaded by the developer and keeps them organized.

Each project has its own node_module folder to keep the data separate

It allows code reusability by giving us access of prebuilt packages etc from the community

We shouldn't push it to github because of its size and also the number of files and folders that it contains.

12) What is the dist folder?

This folder has the final like ready to go/ optimized files are stored after a project is built

Meaning it contains the compiled code which is ready for production or public use.

dist meaning distributable and refers to a directory where files will be stored that can be directly used by others without the need to compile.

13) What is Browserslist ?

Browserslists is a tool that helps which specify which web browsers you want to support in the frontend by using queries in config file

It's like a guide for code to work, making sure the app will work on most browsers and their versions.

It's used by frameworks/libraries such as React, Angular and Vue etc.

In this tool, we can compose the queries by using Query Composition and can use it in our project.

Github Link:

https://github.com/YogiKrishna7/React-Assignment