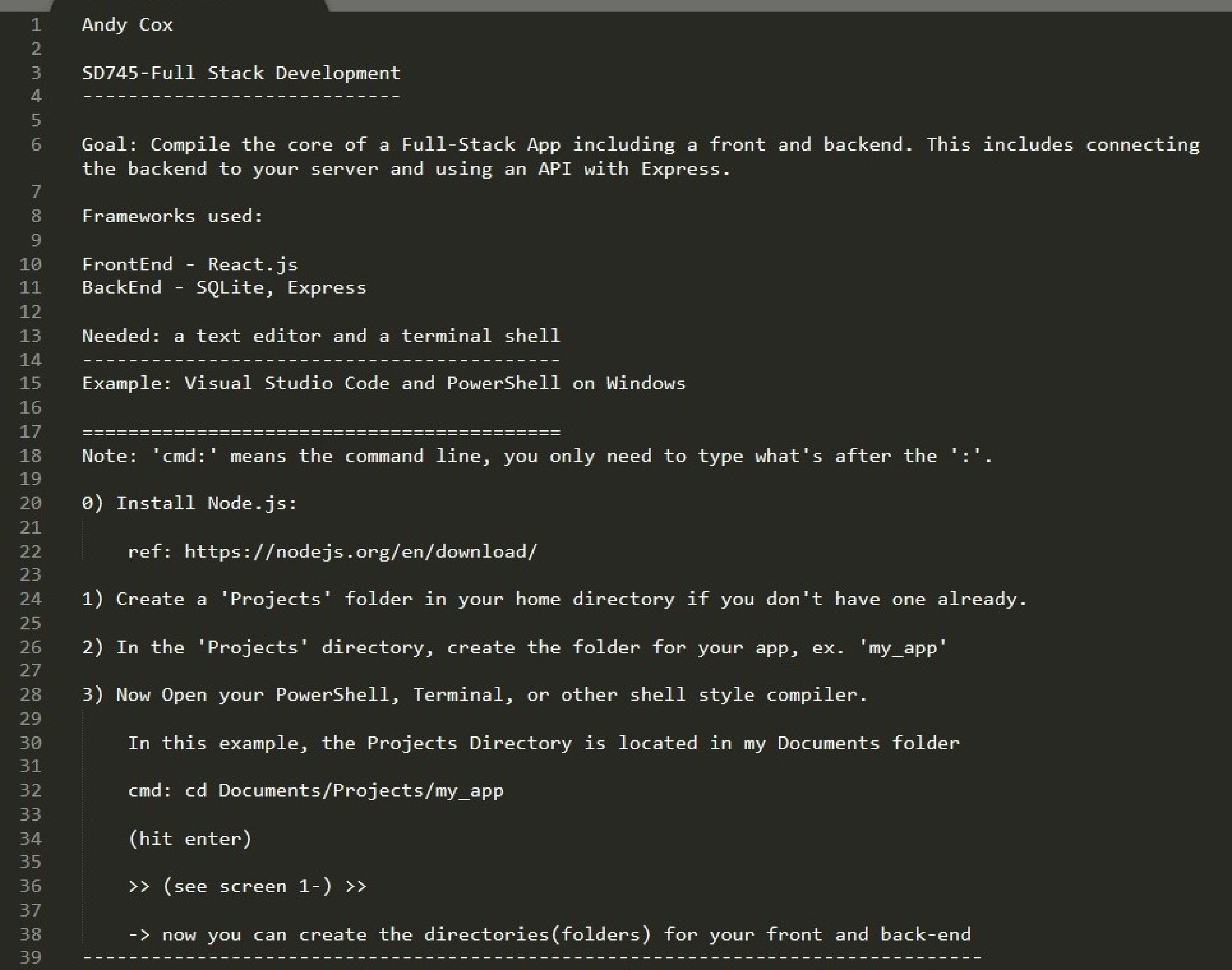
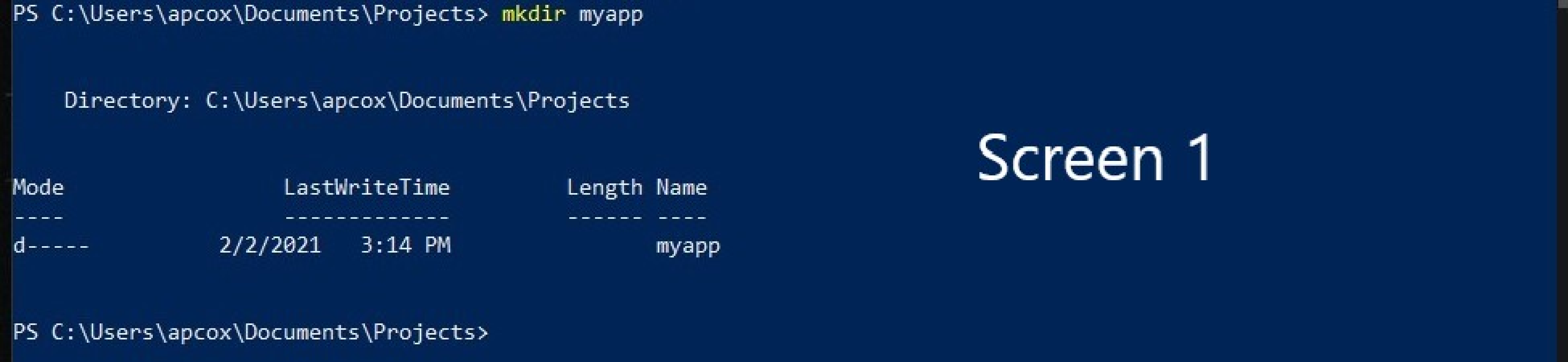
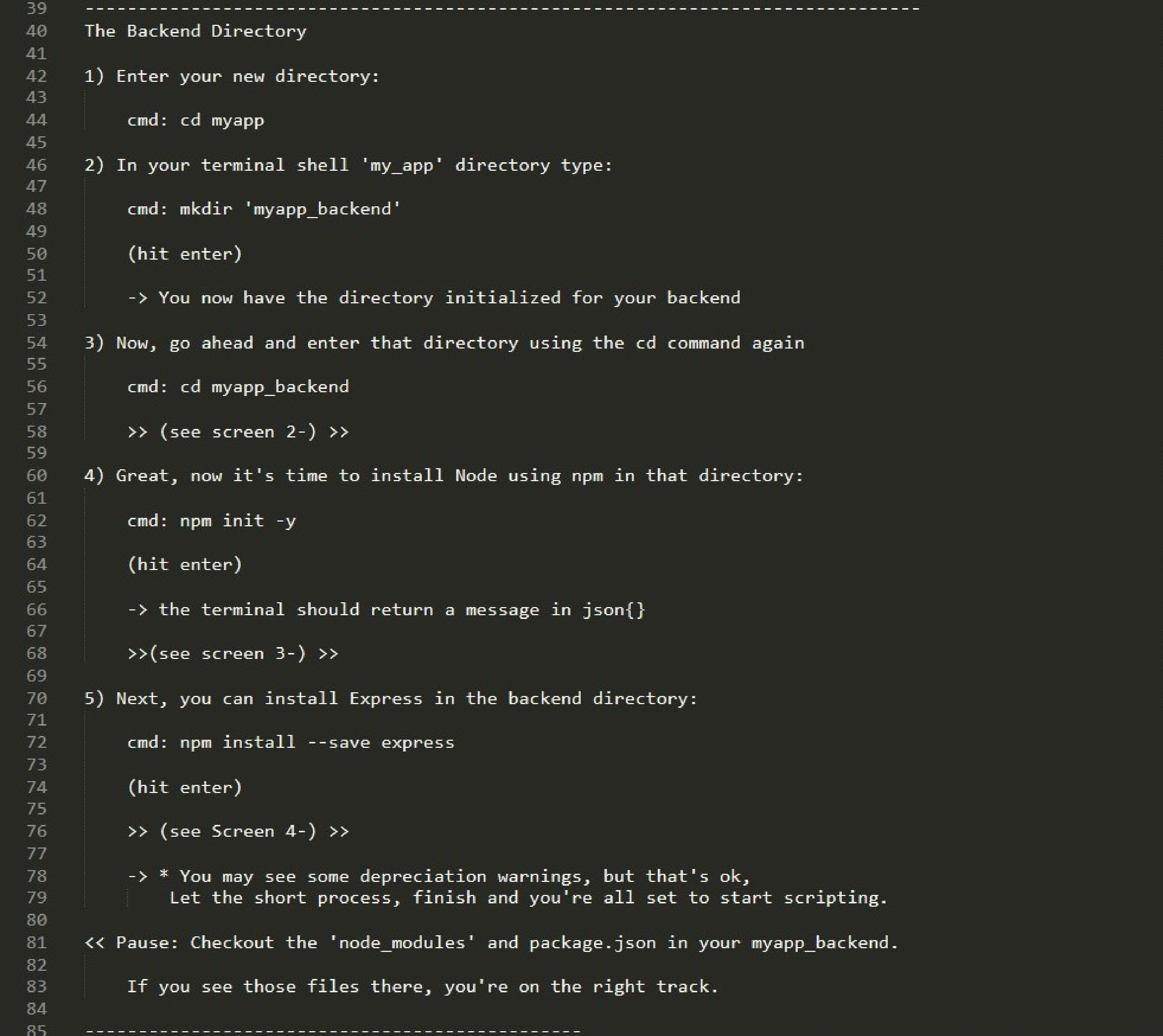


```
Andy Cox

SD745-Full Stack Development
----------------------------

Goal: Compile the core of a Full-Stack App including a front and backend. This includes connecting the backend to your server and using an API with Express.

Frameworks used:

FrontEnd - React.js
BackEnd - SQLite, Express

Needed: a text editor and a terminal shell
------------------------------------------
Example: Visual Studio Code and PowerShell on Windows

==========================================
Note: 'cmd:' means the command line, you only need to type what's after the ':'.

0) Install Node.js:

    ref: https://nodejs.org/en/download/

1) Create a 'Projects' folder in your home directory if you don't have one already.

2) In the 'Projects' directory, create the folder for your app, ex. 'my_app'

3) Now Open your PowerShell, Terminal, or other shell style compiler.

    In this example, the Projects Directory is located in my Documents folder

    cmd: cd Documents/Projects/my_app

    (hit enter)

    >> (see screen 1-) >>

    -> now you can create the directories(folders) for your front and back-end
-------------------------------------------------------------------------------
```

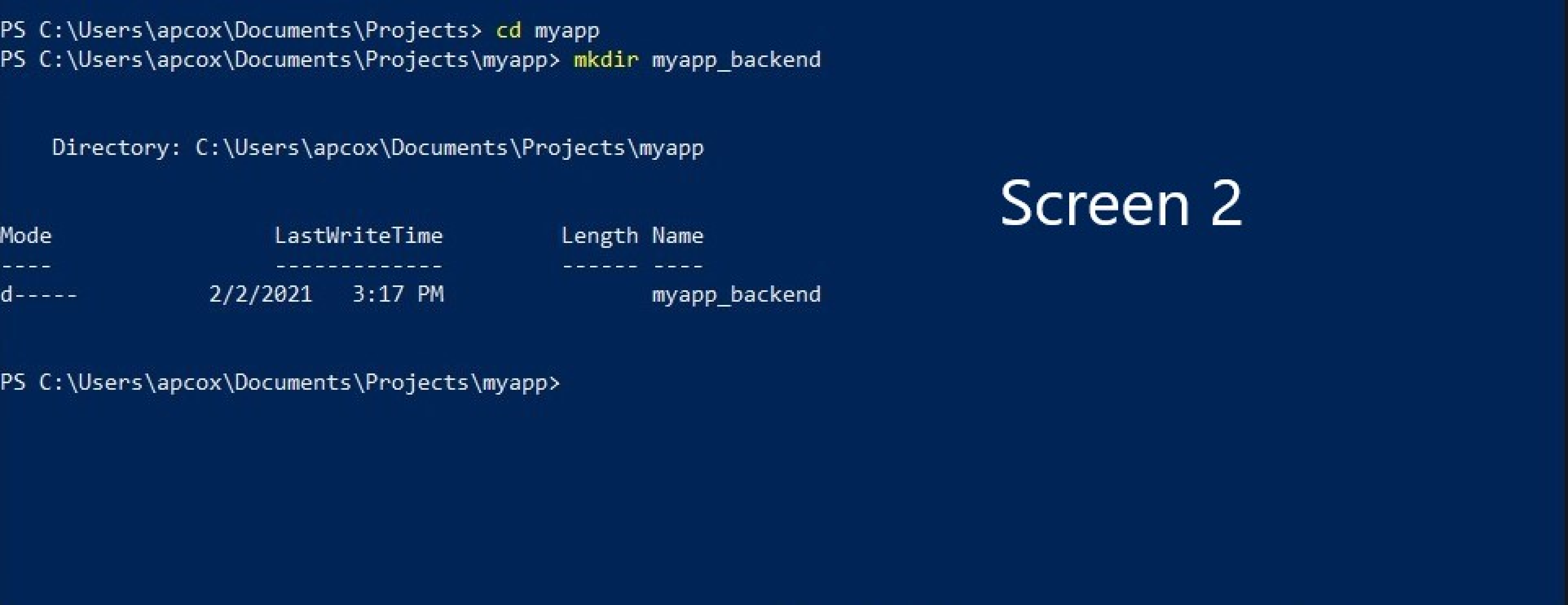


```
PS C:\Users\apcox\Documents\Projects> mkdir myapp


    Directory: C:\Users\apcox\Documents\Projects


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----          2/2/2021   3:14 PM                myapp


PS C:\Users\apcox\Documents\Projects>
```

Screen 1

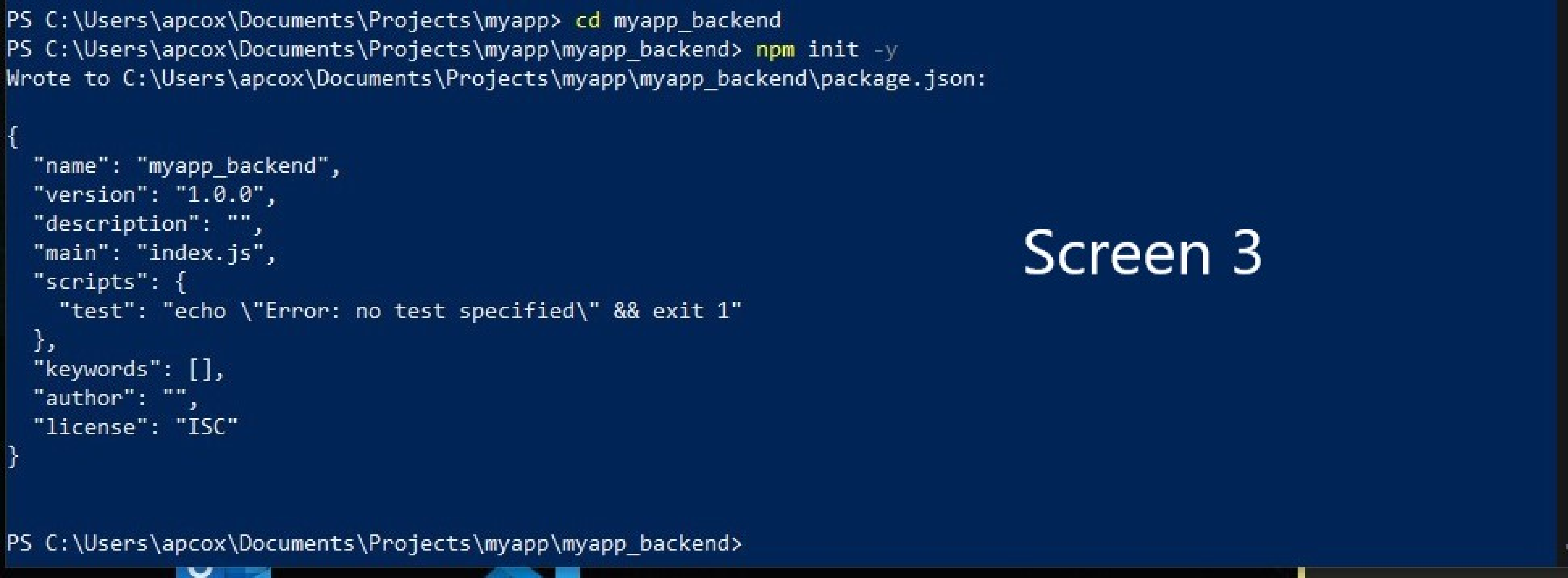


```
--------------------------------------------------------------------------------
The Backend Directory

1) Enter your new directory:

    cmd: cd myapp

2) In your terminal shell 'my_app' directory type:

    cmd: mkdir 'myapp_backend'

    (hit enter)

    -> You now have the directory initialized for your backend

3) Now, go ahead and enter that directory using the cd command again

    cmd: cd myapp_backend

    >> (see screen 2-) >>

4) Great, now it's time to install Node using npm in that directory:

    cmd: npm init -y

    (hit enter)

    -> the terminal should return a message in json{}

    >>(see screen 3-) >>

5) Next, you can install Express in the backend directory:

    cmd: npm install --save express

    (hit enter)

    >> (see Screen 4-) >>

    -> * You may see some depreciation warnings, but that's ok,
        Let the short process, finish and you're all set to start scripting.

<< Pause: Checkout the 'node_modules' and package.json in your myapp_backend.

    If you see those files there, you're on the right track.

--------------------------------------------
```



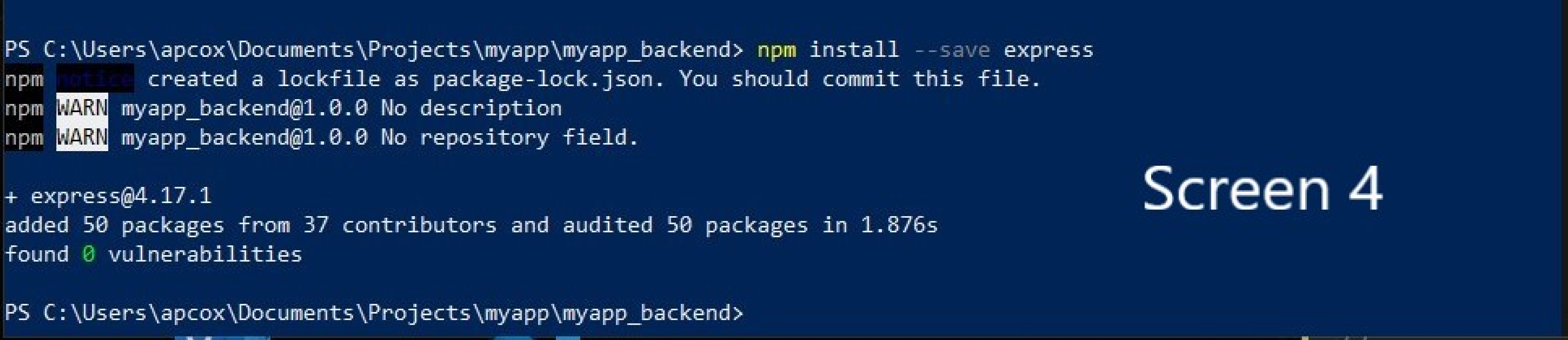
```
PS C:\Users\apcox\Documents\Projects> cd myapp
PS C:\Users\apcox\Documents\Projects\myapp> mkdir myapp_backend


    Directory: C:\Users\apcox\Documents\Projects\myapp


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----          2/2/2021   3:17 PM                myapp_backend


PS C:\Users\apcox\Documents\Projects\myapp>
```

Screen 2



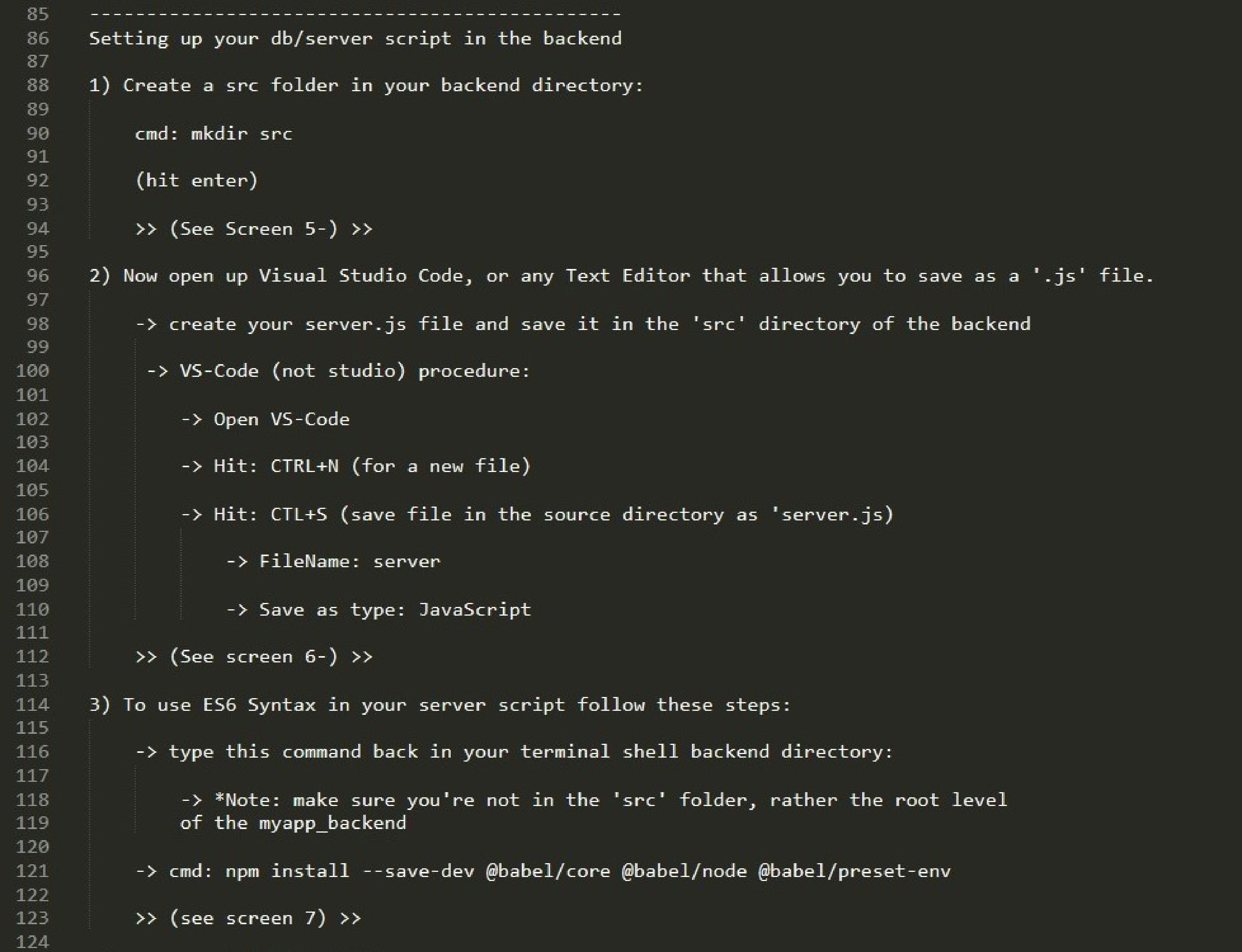
```
PS C:\Users\apcox\Documents\Projects\myapp> cd myapp_backend
PS C:\Users\apcox\Documents\Projects\myapp\myapp_backend> npm init -y
Wrote to C:\Users\apcox\Documents\Projects\myapp\myapp_backend\package.json:

{
  "name": "myapp_backend",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}


PS C:\Users\apcox\Documents\Projects\myapp\myapp_backend>
```

Screen 3

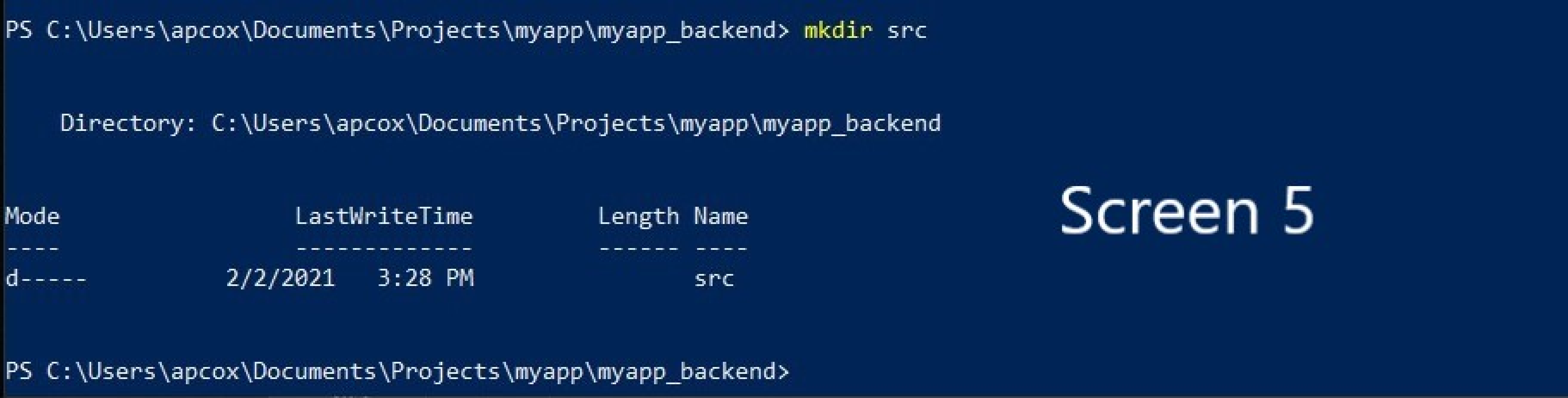


```
PS C:\Users\apcox\Documents\Projects\myapp\myapp_backend> npm install --save express
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN myapp_backend@1.0.0 No description
npm WARN myapp_backend@1.0.0 No repository field.

+ express@4.17.1
added 50 packages from 37 contributors and audited 50 packages in 1.876s
found 0 vulnerabilities

PS C:\Users\apcox\Documents\Projects\myapp\myapp_backend>
```

Screen 4

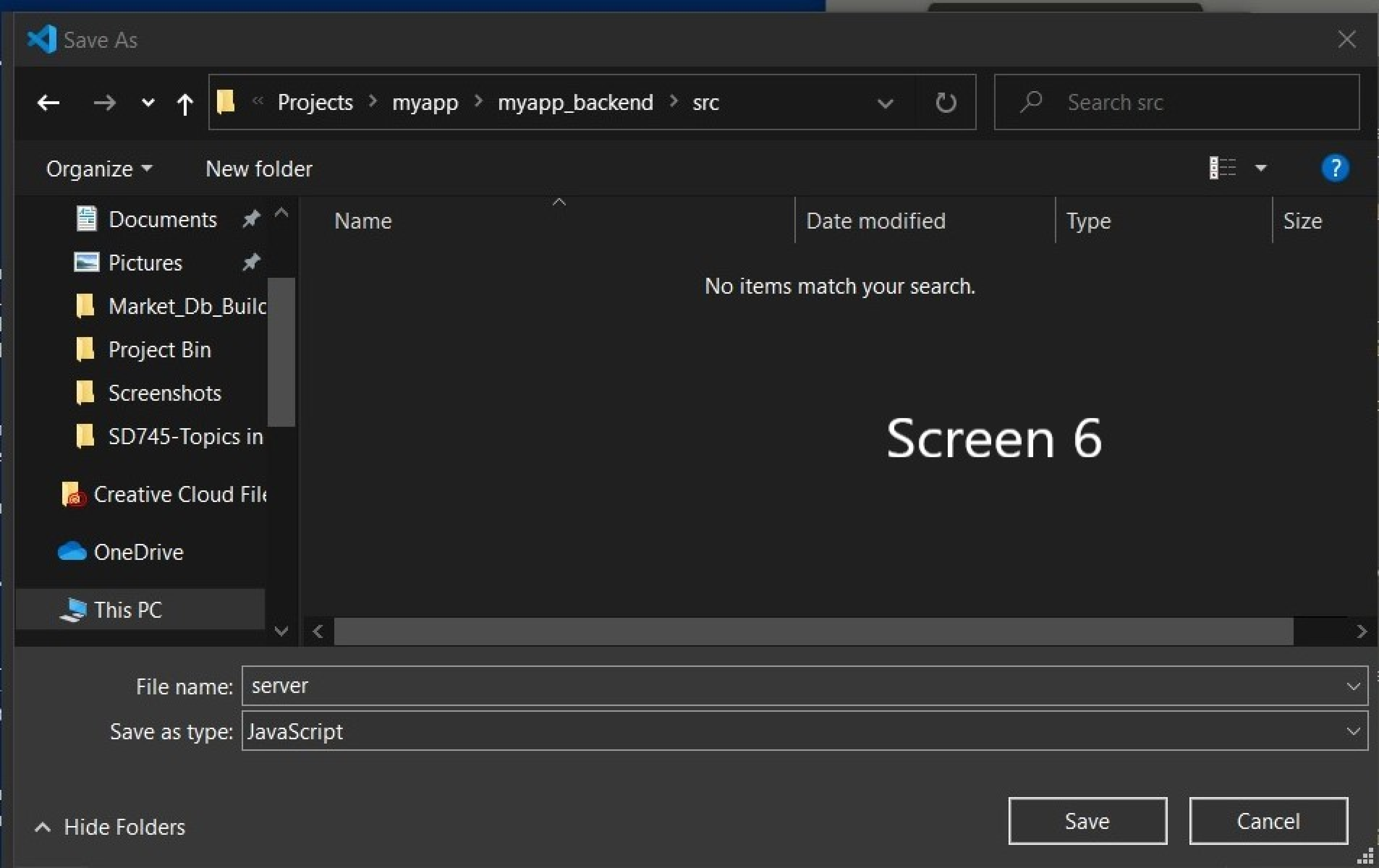


```
-----------------------------------------------
Setting up your db/server script in the backend

1) Create a src folder in your backend directory:

    cmd: mkdir src

    (hit enter)

    >> (See Screen 5-) >>

2) Now open up Visual Studio Code, or any Text Editor that allows you to save as a '.js' file.

    -> create your server.js file and save it in the 'src' directory of the backend

     -> VS-Code (not studio) procedure:

        -> Open VS-Code

        -> Hit: CTRL+N (for a new file)

        -> Hit: CTL+S (save file in the source directory as 'server.js)

            -> FileName: server

            -> Save as type: JavaScript

    >> (See screen 6-) >>

3) To use ES6 Syntax in your server script follow these steps:

    -> type this command back in your terminal shell backend directory:

        -> *Note: make sure you're not in the 'src' folder, rather the root level
        of the myapp_backend

    -> cmd: npm install --save-dev @babel/core @babel/node @babel/preset-env

    >> (see screen 7) >>

```



```
PS C:\Users\apcox\Documents\Projects\myapp\myapp_backend> mkdir src


    Directory: C:\Users\apcox\Documents\Projects\myapp\myapp_backend


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----          2/2/2021   3:28 PM                src


PS C:\Users\apcox\Documents\Projects\myapp\myapp_backend>
```

Screen 5

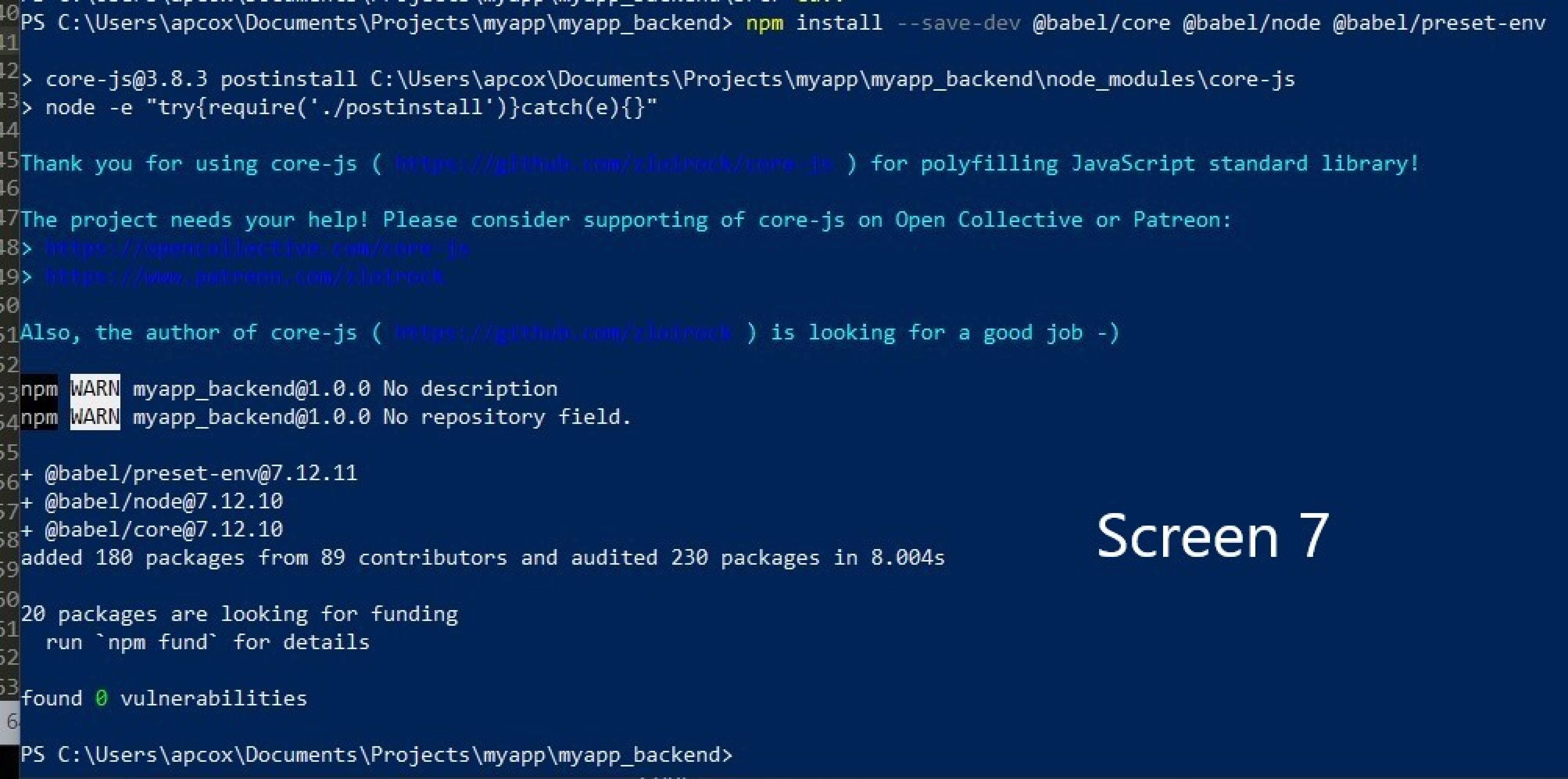


No items match your search.

Screen 6

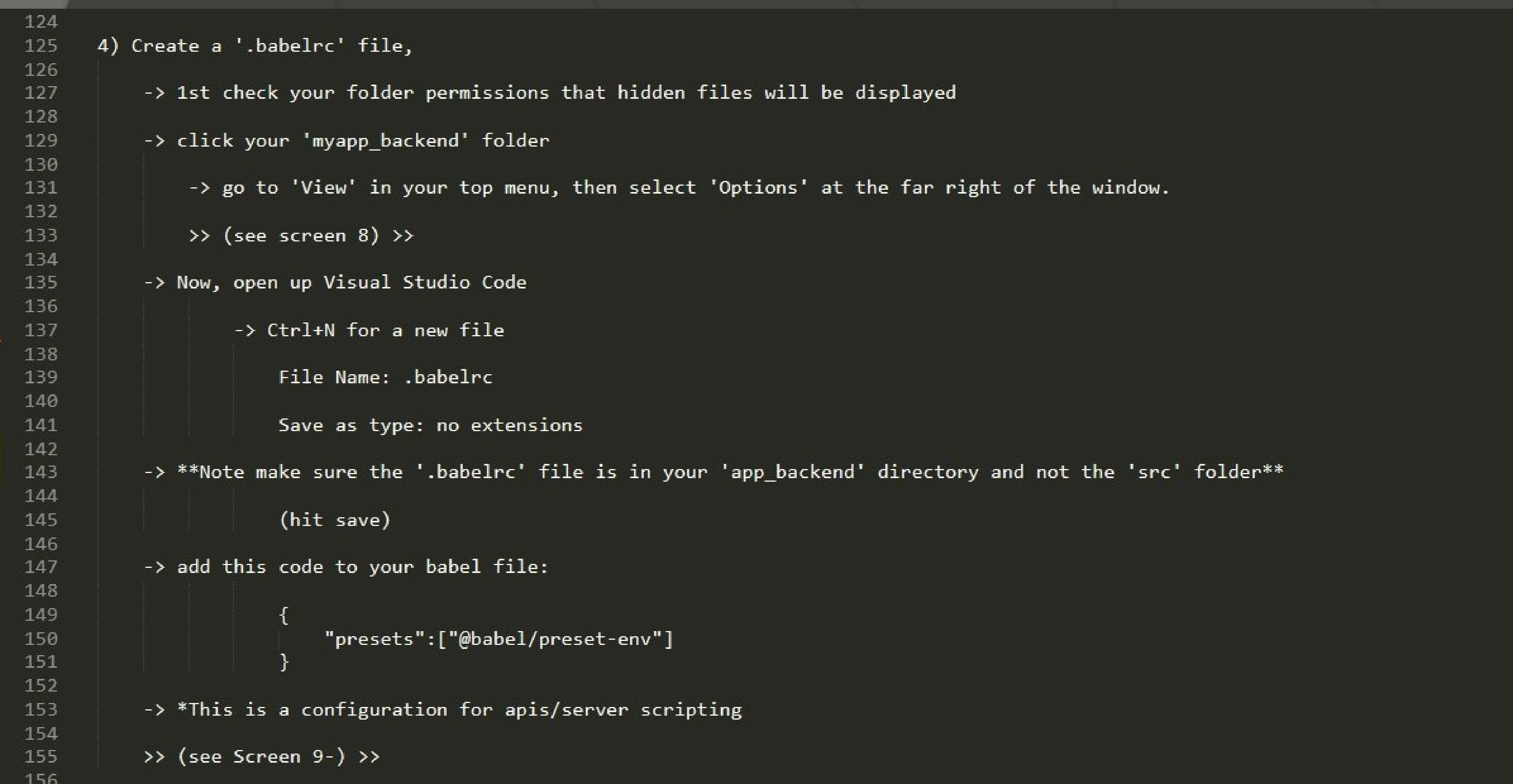
File name: server

Save as type: JavaScript

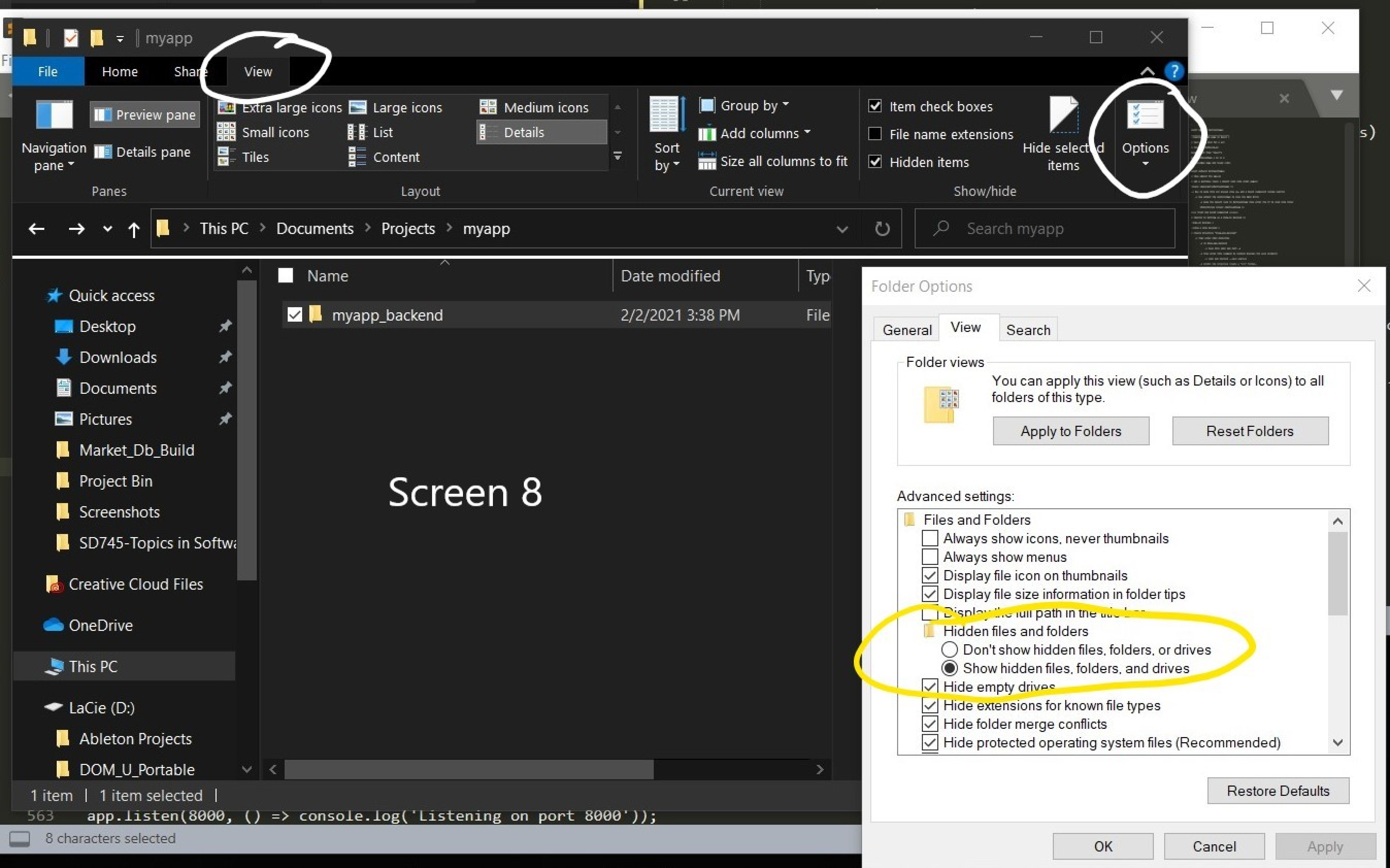


```
PS C:\Users\apcox\Documents\Projects\myapp\myapp_backend> npm install --save-dev @babel/core @babel/node @babel/preset-env

> core-js@3.8.3 postinstall C:\Users\apcox\Documents\Projects\myapp\myapp_backend\node_modules\core-js
> node -e "try{require('./postinstall')}catch(e){}"

Thank you for using core-js ( https://github.com/zloirock/core-js ) for polyfilling JavaScript standard library!

The project needs your help! Please consider supporting of core-js on Open Collective or Patreon:
> https://opencollective.com/core-js
> https://www.patreon.com/zloirock

Also, the author of core-js ( https://github.com/zloirock ) is looking for a good job -)

npm WARN myapp_backend@1.0.0 No description
npm WARN myapp_backend@1.0.0 No repository field.

+ @babel/preset-env@7.12.11
+ @babel/node@7.12.10
+ @babel/core@7.12.10
added 180 packages from 89 contributors and audited 230 packages in 8.004s

20 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

PS C:\Users\apcox\Documents\Projects\myapp\myapp_backend>
```

Screen 7



```
4) Create a '.babelrc' file,

    -> 1st check your folder permissions that hidden files will be displayed

    -> click your 'myapp_backend' folder

        -> go to 'View' in your top menu, then select 'Options' at the far right of the window.

        >> (see screen 8) >>

    -> Now, open up Visual Studio Code

            -> Ctrl+N for a new file

                File Name: .babelrc

                Save as type: no extensions

    -> **Note make sure the '.babelrc' file is in your 'app_backend' directory and not the 'src' folder**

                (hit save)

    -> add this code to your babel file:

                {
                    "presets":["@babel/preset-env"]
                }

    -> *This is a configuration for apis/server scripting

    >> (see Screen 9-) >>
```



☑ Item check boxes
☐ File name extensions
☑ Hidden items

☐ Name	Date modified	Typ
☑ myapp_backend	2/2/2021 3:38 PM	File

Screen 8

1 item | 1 item selected |

```
563    app.listen(8000, () => console.log('Listening on port 8000'));
```

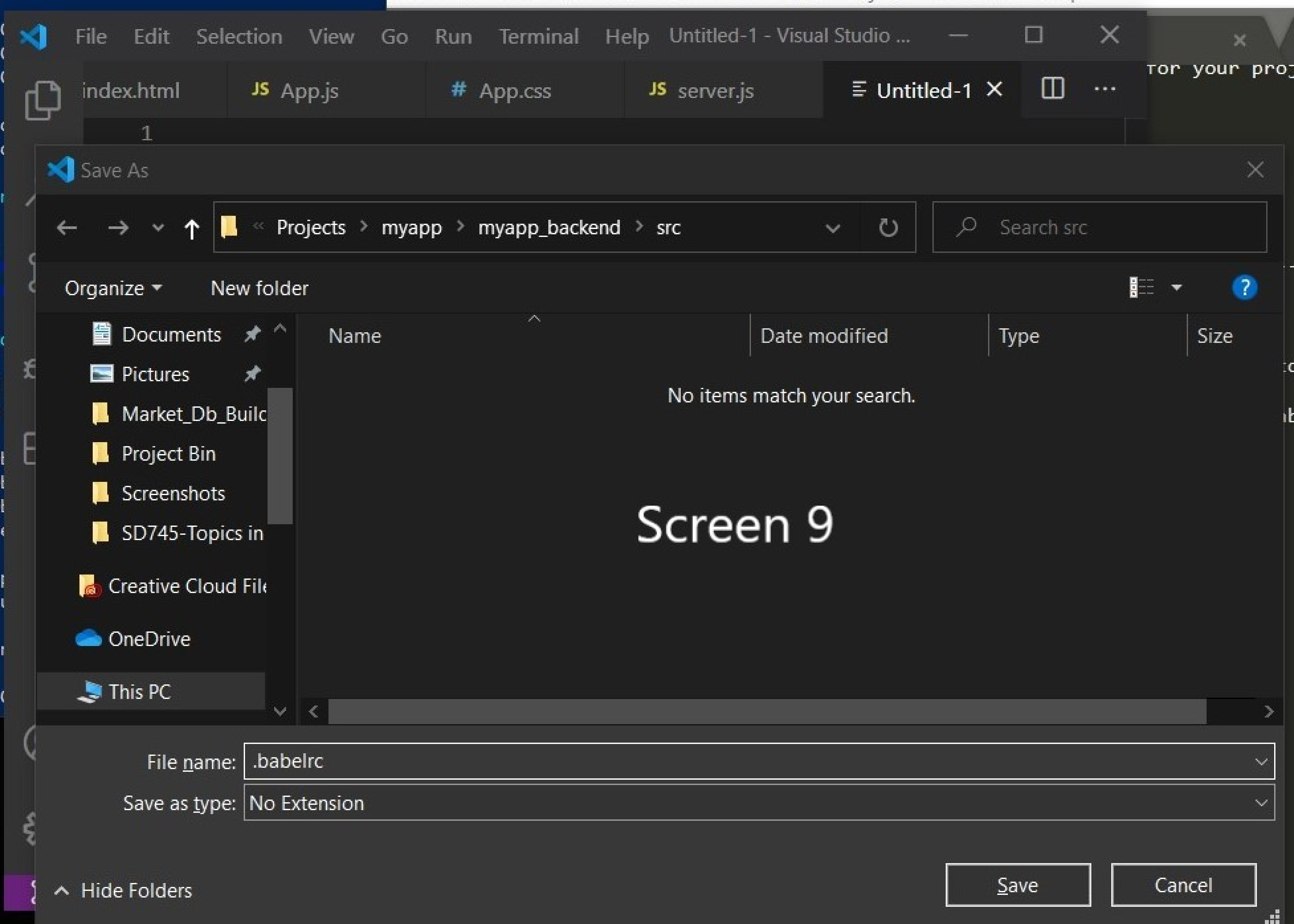
8 characters selected

Folder views

You can apply this view (such as Details or Icons) to all folders of this type.

Advanced settings:

Files and Folders
- ☐ Always show icons, never thumbnails
- ☐ Always show menus
- ☑ Display file icon on thumbnails
- ☑ Display file size information in folder tips
- ☐ Display the full path in the title bar
- Hidden files and folders
  - ○ Don't show hidden files, folders, or drives
  - ● Show hidden files, folders, and drives
- ☑ Hide empty drives
- ☑ Hide extensions for known file types
- ☑ Hide folder merge conflicts
- ☑ Hide protected operating system files (Recommended)



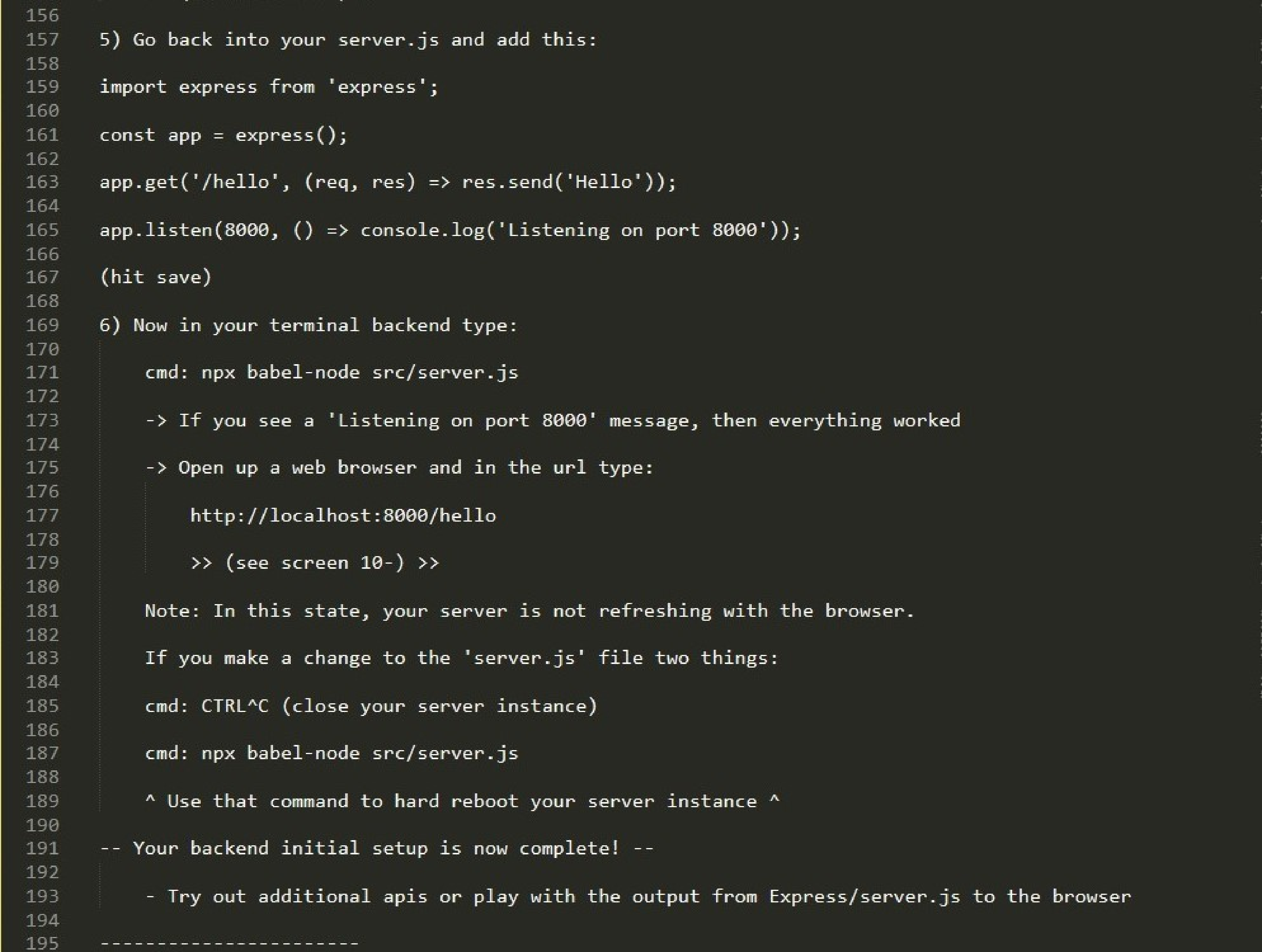
Projects › myapp › myapp_backend › src

No items match your search.

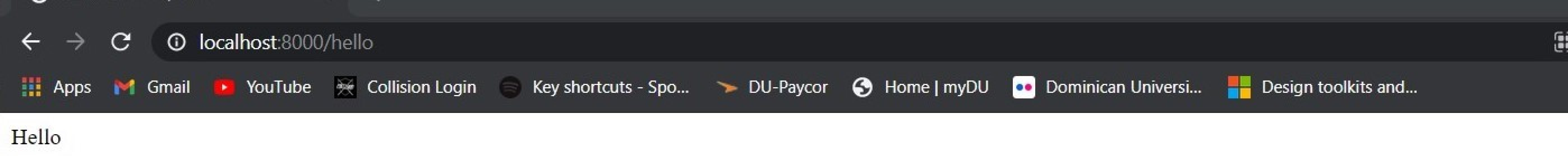
Screen 9

File name: .babelrc

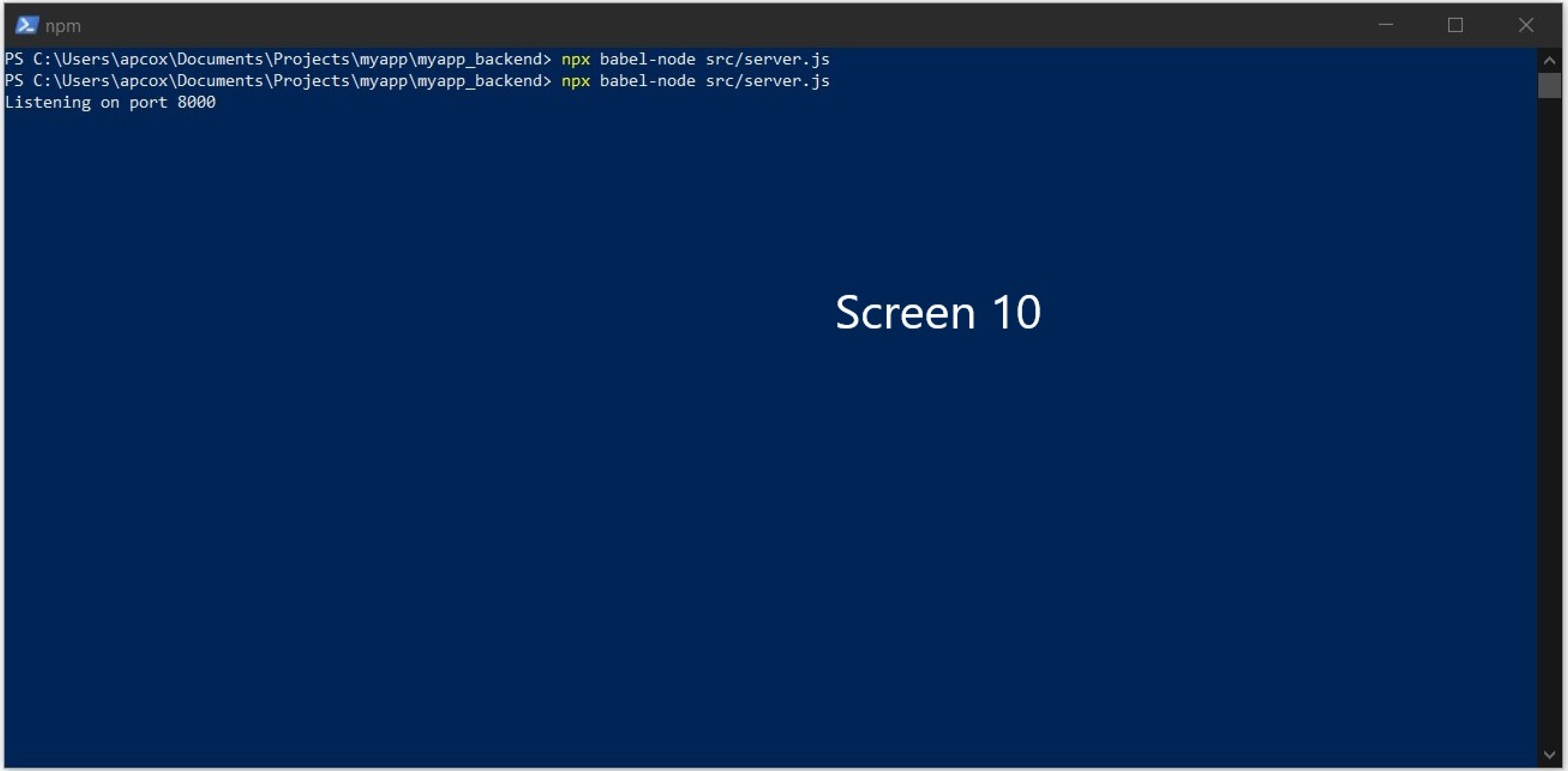
Save as type: No Extension



```
5) Go back into your server.js and add this:

import express from 'express';

const app = express();

app.get('/hello', (req, res) => res.send('Hello'));

app.listen(8000, () => console.log('Listening on port 8000'));

(hit save)

6) Now in your terminal backend type:

    cmd: npx babel-node src/server.js

    -> If you see a 'Listening on port 8000' message, then everything worked

    -> Open up a web browser and in the url type:

        http://localhost:8000/hello

        >> (see screen 10-) >>

    Note: In this state, your server is not refreshing with the browser.

    If you make a change to the 'server.js' file two things:

    cmd: CTRL^C (close your server instance)

    cmd: npx babel-node src/server.js

    ^ Use that command to hard reboot your server instance ^

-- Your backend initial setup is now complete! --

    - Try out additional apis or play with the output from Express/server.js to the browser

-----------------------
```

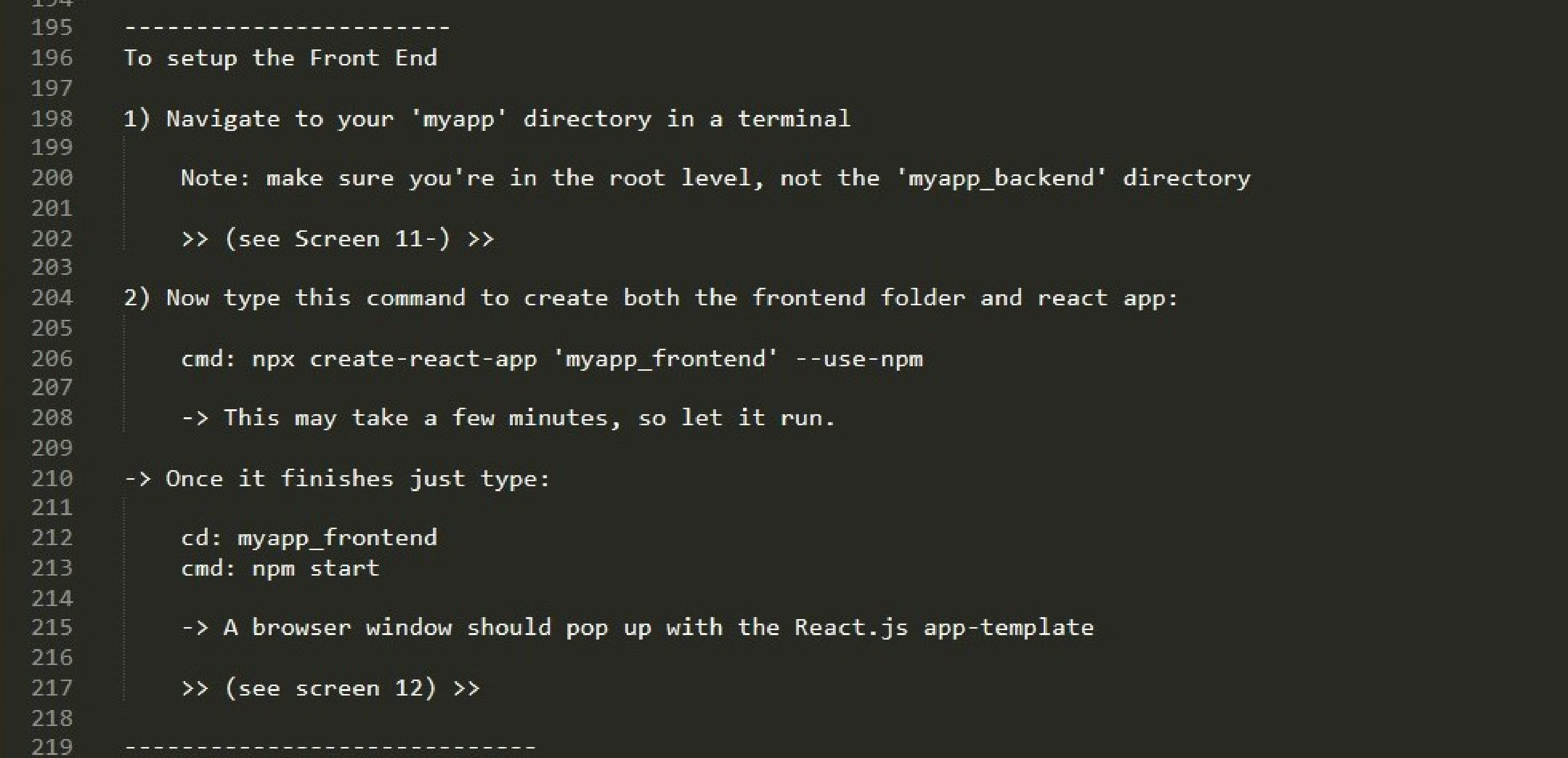


```
Hello
```

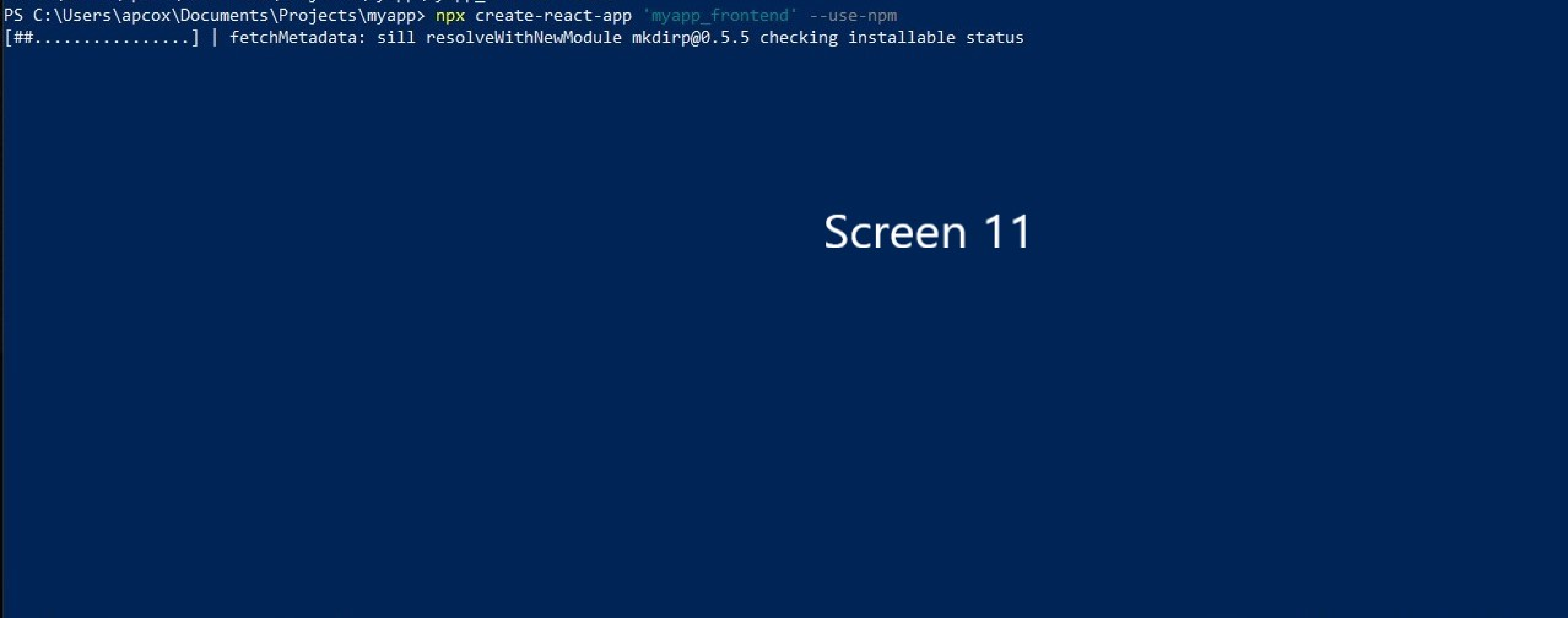


```
PS C:\Users\apcox\Documents\Projects\myapp\myapp_backend> npx babel-node src/server.js
PS C:\Users\apcox\Documents\Projects\myapp\myapp_backend> npx babel-node src/server.js
Listening on port 8000
```

Screen 10

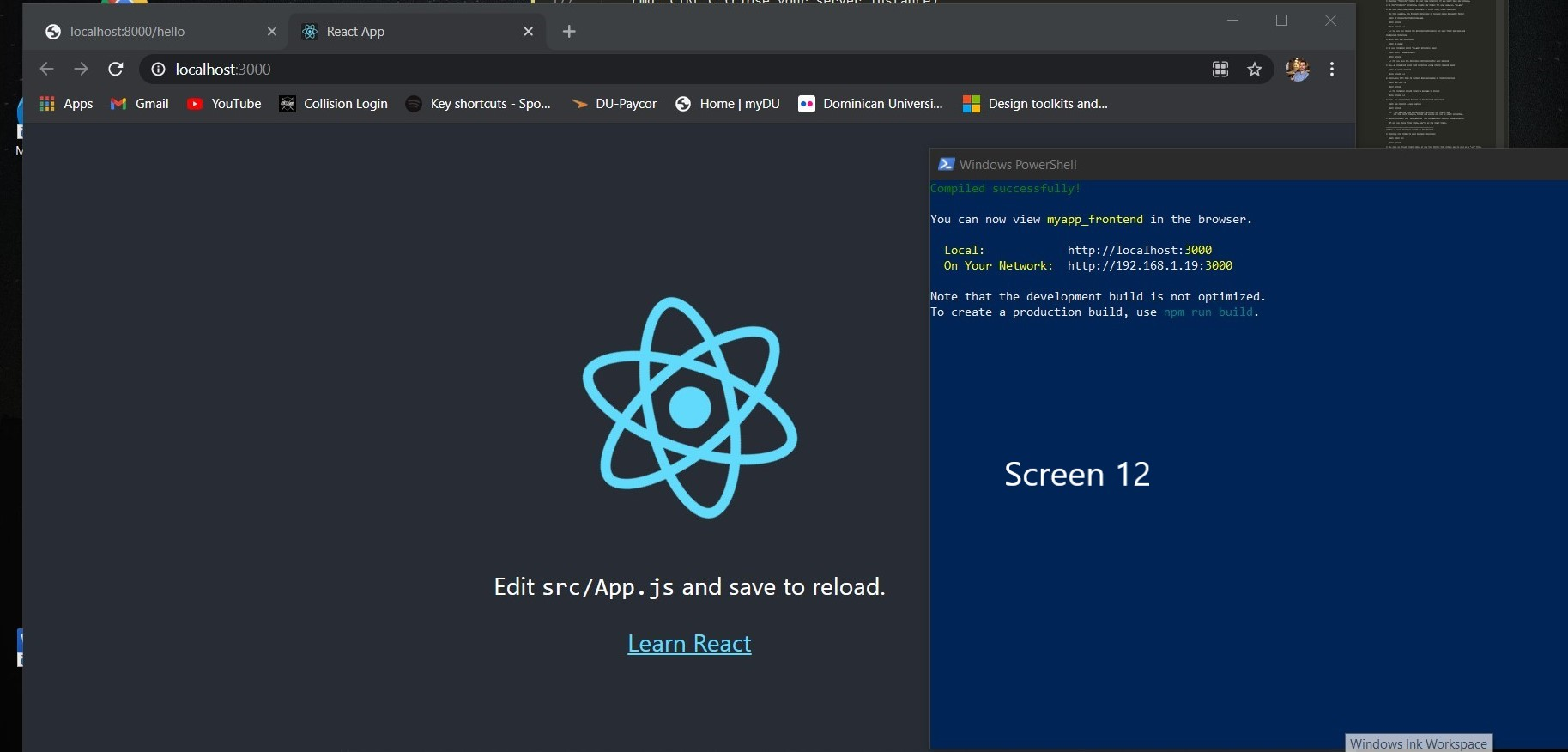


```
-----------------------
To setup the Front End

1) Navigate to your 'myapp' directory in a terminal

    Note: make sure you're in the root level, not the 'myapp_backend' directory

    >> (see Screen 11-) >>

2) Now type this command to create both the frontend folder and react app:

    cmd: npx create-react-app 'myapp_frontend' --use-npm

    -> This may take a few minutes, so let it run.

-> Once it finishes just type:

    cd: myapp_frontend
    cmd: npm start

    -> A browser window should pop up with the React.js app-template

    >> (see screen 12) >>

-----------------------------
```



```
PS C:\Users\apcox\Documents\Projects\myapp> npx create-react-app 'myapp_frontend' --use-npm
[##...............] | fetchMetadata: sill resolveWithNewModule mkdirp@0.5.5 checking installable status
```

Screen 11

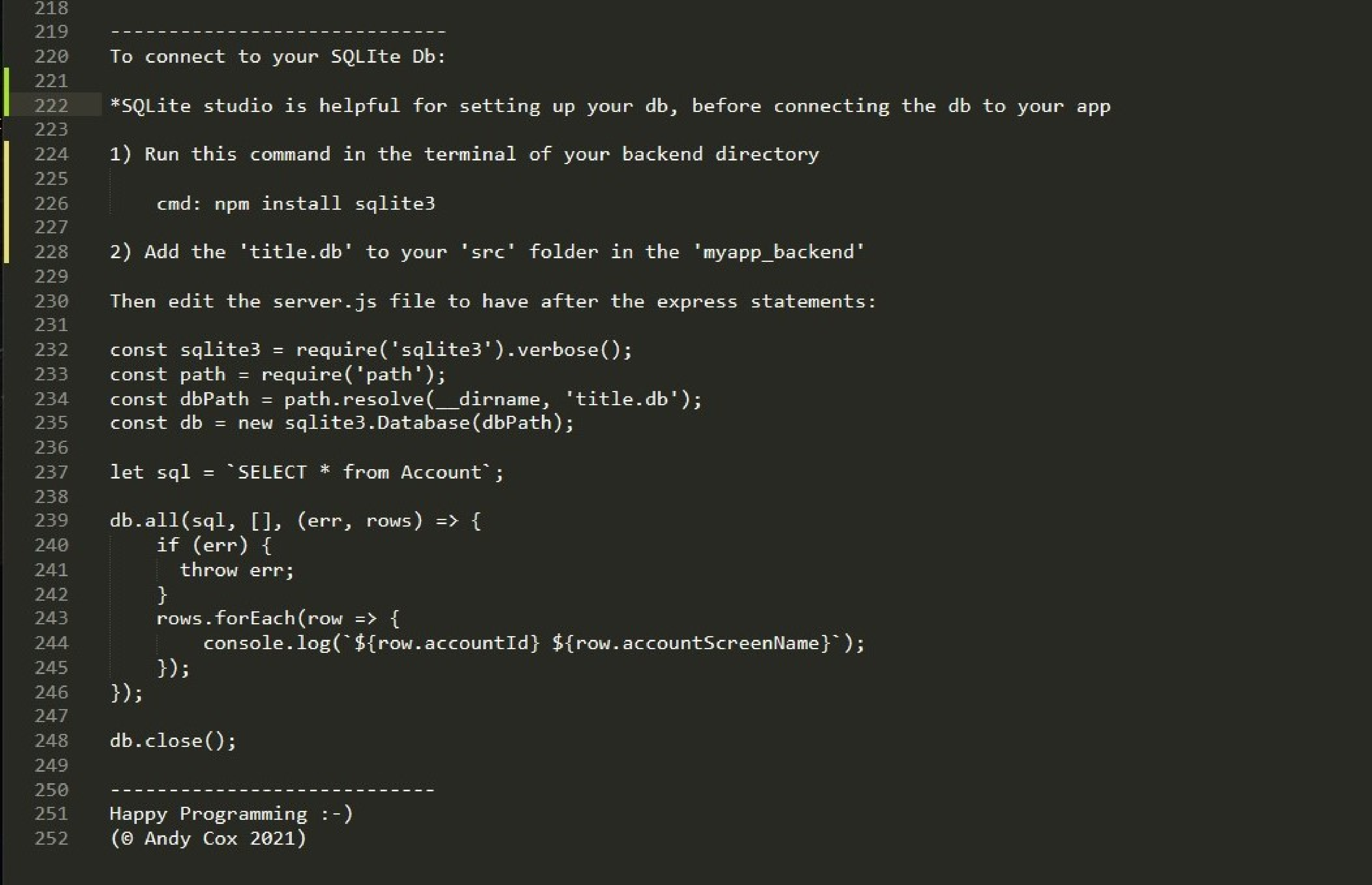


Edit `src/App.js` and save to reload.

Learn React

```
Compiled successfully!

You can now view myapp_frontend in the browser.

  Local:            http://localhost:3000
  On Your Network:  http://192.168.1.19:3000

Note that the development build is not optimized.
To create a production build, use npm run build.
```

Screen 12



To connect to your SQLIte Db:

*SQLite studio is helpful for setting up your db, before connecting the db to your app

1) Run this command in the terminal of your backend directory

```
    cmd: npm install sqlite3
```

2) Add the 'title.db' to your 'src' folder in the 'myapp_backend'

Then edit the server.js file to have after the express statements:

```
const sqlite3 = require('sqlite3').verbose();
const path = require('path');
const dbPath = path.resolve(__dirname, 'title.db');
const db = new sqlite3.Database(dbPath);

let sql = `SELECT * from Account`;

db.all(sql, [], (err, rows) => {
    if (err) {
      throw err;
    }
    rows.forEach(row => {
        console.log(`${row.accountId} ${row.accountScreenName}`);
    });
});

db.close();
```

Happy Programming :-)