

# REACT/ASP.NET STARTER SETUP GUIDE

## Tech Stack

React.js, HTML/CSS, JavaScript, C#, ASP.NET, Postman, JSON Web Token, React Router

## Notes

This guide will walk you through the process of setting up the devCodeCamp React/ASP.NET starter code, provided for the **BookNook** and **Capstone** projects.

💡 Do not modify the **CarsController** file - this is for demonstration purposes only!

## Setup

### Setting Up the ASP.NET Backend

1. Download the zip file from the Repo:
  - a. **Backend Starter Code Repo**
2. Extract the zipfile into a folder.
3. Change the name of the external folder to a name more appropriate for the project you are building. **Do not change the name of any other internal folders or files – this will cause a cascade of namespace issues!**
4. Change the values in the “**DefaultConnection**” setting of the **appsettings.json** file. Replace **<db\_name>** with the name you want to use for your database (ie **booknook\_webapi**, etc.). Change **<password>** to your MySQL

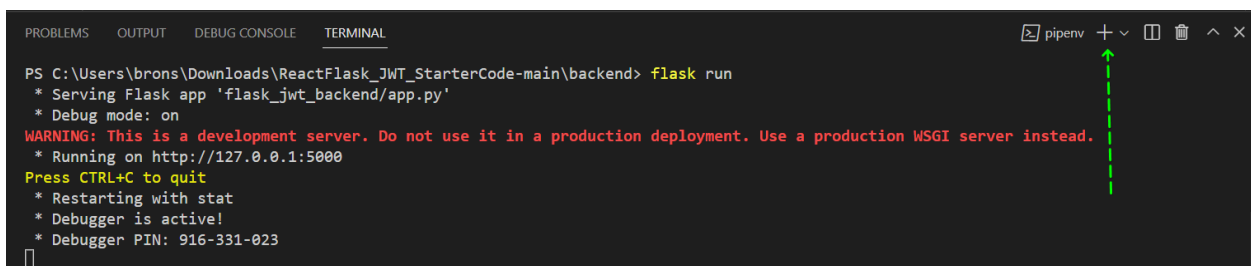
password.

```
"ConnectionStrings": {  
  "DefaultConnection":  
    "Server=localhost;Port=3306;Database=<db_name>;User=root;Password=<password>;"  
},  
"Settings": {  
  "JWT": {  
    "Secret": "secret"  
  }  
}
```

5. Update the database with the included initial migration:
  - a. Windows Users:
    - i. Open the Package Manager Console (Tools > NuGet Package Manager > Package Manager Console).
    - ii. Apply the migration to your database with the command:  
**Update-Database**
  - b. Mac Users:
    - i. Open a terminal and navigate it to the location of your .sln file.
    - ii. Apply the migration to your database with the command:  
**dotnet ef database update**
6. Setup is now complete! Next steps:
  - a. Add your Model classes (and any necessary DTOs)
  - b. Create your DbSet within the ApplicationDbContext.
  - c. Remember to add a new migration and update the database after creating (or modifying) models.
  - d. Create your Controllers
  - e. Work on your endpoints one at a time, testing in Postman!

## Setting Up the React Frontend

1. Download the zip file from the Repo:
  - a. [Frontend Starter Code Repo](#)
2. Create a new Instance of a Terminal in Visual Studio Code



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL  
PS C:\Users\brons\Downloads\ReactFlask_JWT_StarterCode-main\backend> flask run  
* Serving Flask app 'flask_jwt_backend/app.py'  
* Debug mode: on  
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.  
* Running on http://127.0.0.1:5000  
Press CTRL+C to quit  
* Restarting with stat  
* Debugger is active!  
* Debugger PIN: 916-331-023
```

3. Your new Terminal instance will start at the parent directory of the project. Change the directory into the **frontend** directory using **cd frontend**.
4. Run the **npm install** command to install React dependencies
5. Run the **npm start** command to start your React application!

```
Compiled successfully!

You can now view frontend in the browser.

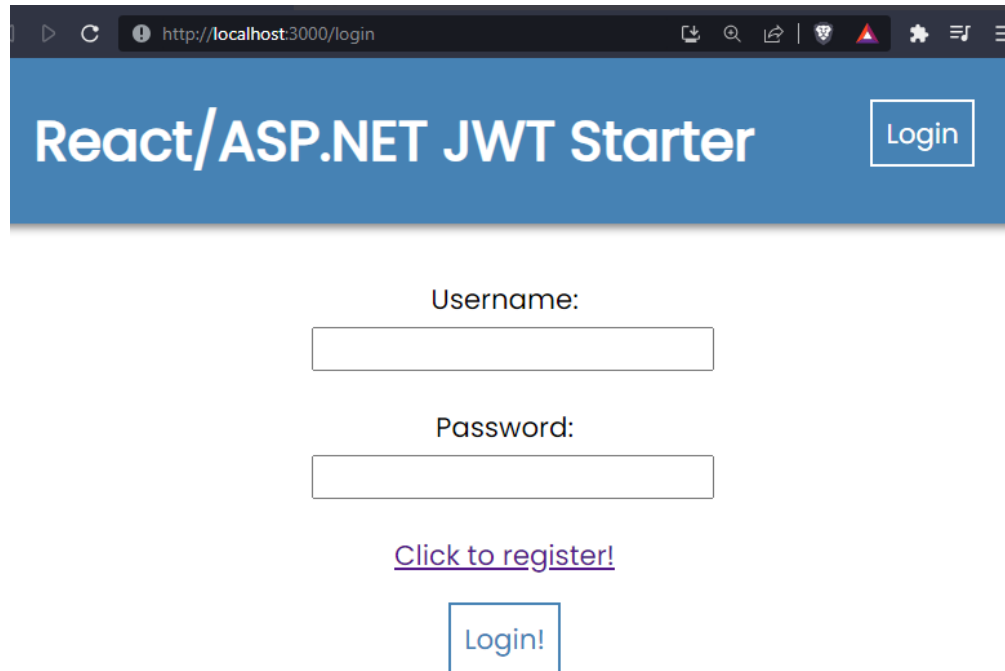
  Local:            http://localhost:3000
  On Your Network:  http://192.168.86.24:3000

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

asset static/js/bundle.js 1.66 MiB [emitted] (name: main) 1 related asset
asset index.html 1.67 KiB [emitted]
asset asset-manifest.json 190 bytes [emitted]
cached modules 1.49 MiB (javascript) 28.1 KiB (runtime) [cached] 164 modules
webpack 5.69.0 compiled successfully in 1560 ms
```

*The React server running successfully*

6. Your React application should open in your browser and automatically route to the Login screen.



React/ASP.NET JWT Starter

Login

Username:

Password:

[Click to register!](#)

Login!

## Common Problems and Reminders

- If you see the below message, make sure your Application's Local Storage is clear!
  - This article links to how to do this in Google Chrome and will work similarly in other browsers:  
<https://www.leadshook.com/help/how-to-clear-local-storage-in-google-chrome-browser/>

```
✖ ▶ Uncaught SyntaxError: Unexpected token e in JSON at position 0
    at JSON.parse (<anonymous>)
    at AuthProvider (bundle.js:sourceMap:401:26)
    at renderWithHooks (bundle.js:sourceMap:26155:22)
    at mountIndeterminateComponent (bundle.js:sourceMap:28917:17)
    at beginWork (bundle.js:sourceMap:30116:20)
    at HTMLUnknownElement.callCallback (bundle.js:sourceMap:15105:18)
    at Object.invokeGuardedCallbackDev (bundle.js:sourceMap:15154:20)
    at invokeGuardedCallback (bundle.js:sourceMap:15214:35)
    at beginWork$1 (bundle.js:sourceMap:34956:11)
    at performUnitOfWork (bundle.js:sourceMap:33792:16)

✖ ▶ The above error occurred in the <AuthProvider> component:

    at AuthProvider (http://localhost:3000/static/js/bundle.js:398:5)
    at Router (http://localhost:3000/static/js/bundle.js:38620:15)
    at BrowserRouter (http://localhost:3000/static/js/bundle.js:38100:5)

Consider adding an error boundary to your tree to customize error handling behavior.
Visit https://reactjs.org/link/error-boundaries to learn more about error boundaries.

✖ ▶ Uncaught SyntaxError: Unexpected token e in JSON at position 0
    at JSON.parse (<anonymous>)
    at AuthProvider (bundle.js:sourceMap:401:26)
    at renderWithHooks (bundle.js:sourceMap:26155:22)
    at mountIndeterminateComponent (bundle.js:sourceMap:28917:17)
    at beginWork (bundle.js:sourceMap:30116:20)
    at HTMLUnknownElement.callCallback (bundle.js:sourceMap:15105:18)
    at Object.invokeGuardedCallbackDev (bundle.js:sourceMap:15154:20)
    at invokeGuardedCallback (bundle.js:sourceMap:15214:35)
    at beginWork$1 (bundle.js:sourceMap:34956:11)
    at performUnitOfWork (bundle.js:sourceMap:33792:16)
```