Bucket List Tracker

Create and Deploy React Frontend App:

AWS Amplify provides a Git-based CI/CD workflow that allows to build, deploy, and host single-page web applications or static sites with serverless backends.

It automatically detects the build settings for both the frontend and any serverless backend resources when connected to a Git repository. With each code commit, Amplify redeploys updates automatically.

In this section, I'll create a new React application for bucket list tracker, push it to a GitHub repository, and connect it to AWS Amplify for deployment.

Create a React Application:

1.Open a new terminal window and run the following commands to create a React app using Vite:

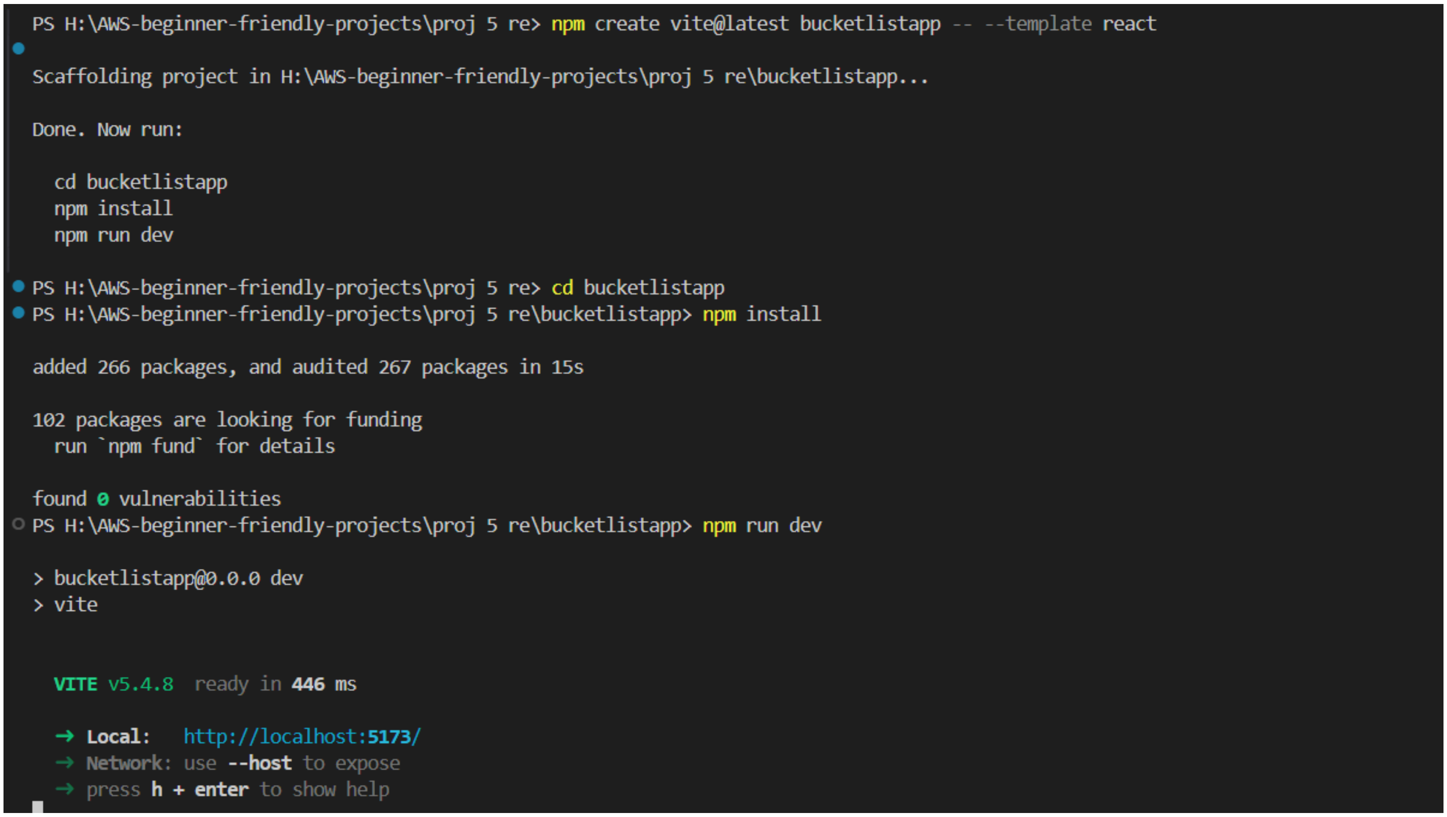
npm create vite@latest bucketlistapp -- --template react

cd bucketlistapp

npm install

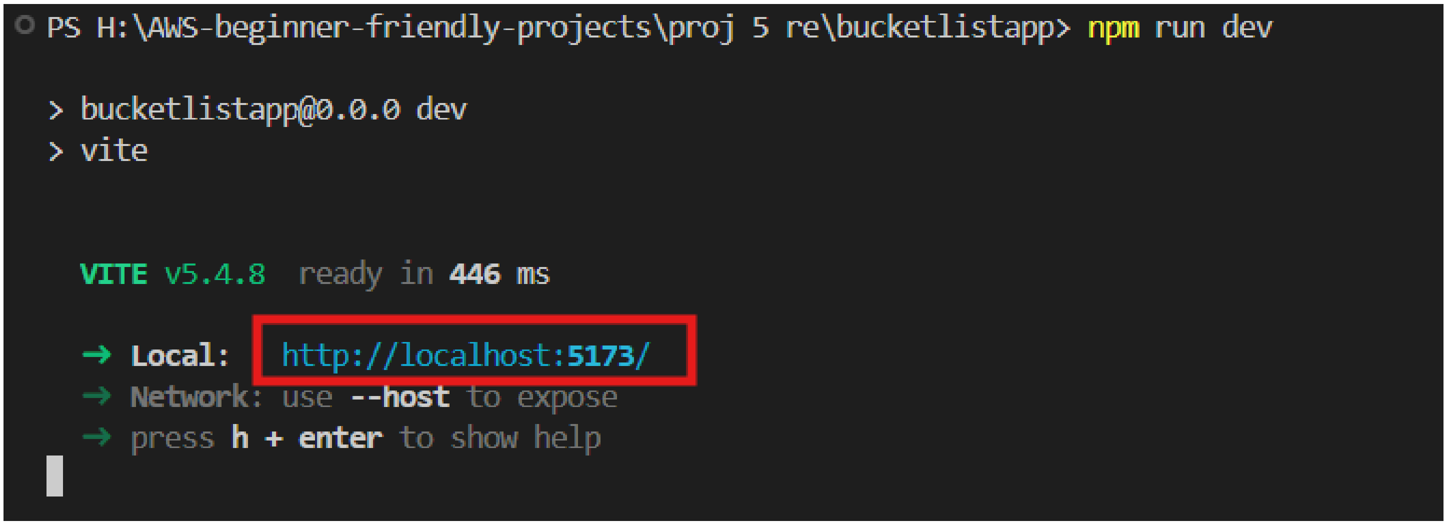
npm run dev

You can check the below screenshot for the flow of the commands:



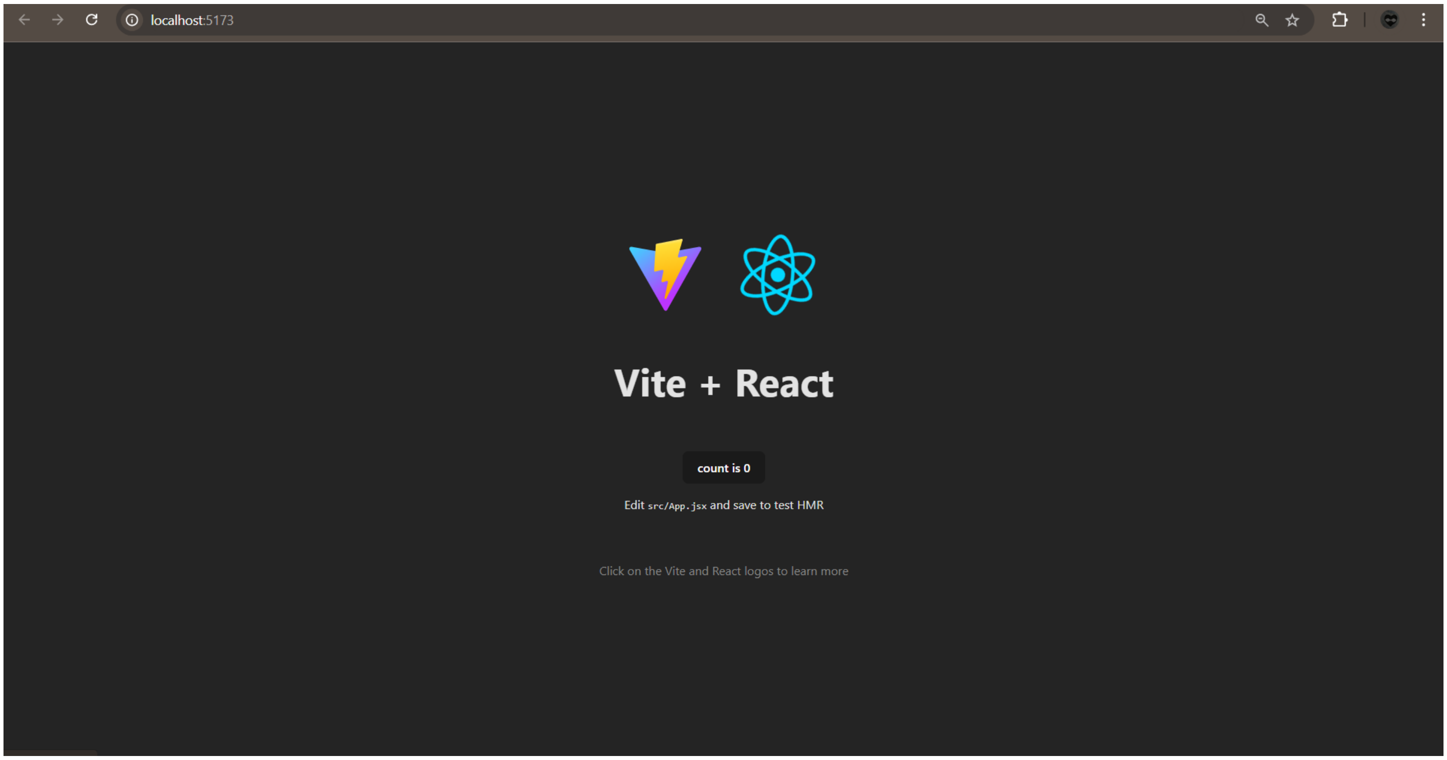
```
PS H:\AWS-beginner-friendly-projects\proj 5 re> npm create vite@latest bucketlistapp -- --template react

Scaffolding project in H:\AWS-beginner-friendly-projects\proj 5 re\bucketlistapp...

Done. Now run:

  cd bucketlistapp
  npm install
  npm run dev

PS H:\AWS-beginner-friendly-projects\proj 5 re> cd bucketlistapp
PS H:\AWS-beginner-friendly-projects\proj 5 re\bucketlistapp> npm install

added 266 packages, and audited 267 packages in 15s

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

found 0 vulnerabilities
PS H:\AWS-beginner-friendly-projects\proj 5 re\bucketlistapp> npm run dev

> bucketlistapp@0.0.0 dev
> vite


  VITE v5.4.8  ready in 446 ms

  ➜  Local:   http://localhost:5173/
  ➜  Network: use --host to expose
  ➜  press h + enter to show help
```

2. In the terminal window, click the Local link to open the Vite + React application in browser and view bucket list tracker.



```
PS H:\AWS-beginner-friendly-projects\proj 5 re\bucketlistapp> npm run dev

> bucketlistapp@0.0.0 dev
> vite


  VITE v5.4.8  ready in 446 ms

  ➜  Local:   http://localhost:5173/
  ➜  Network: use --host to expose
  ➜  press h + enter to show help
```

successfully created a React application.



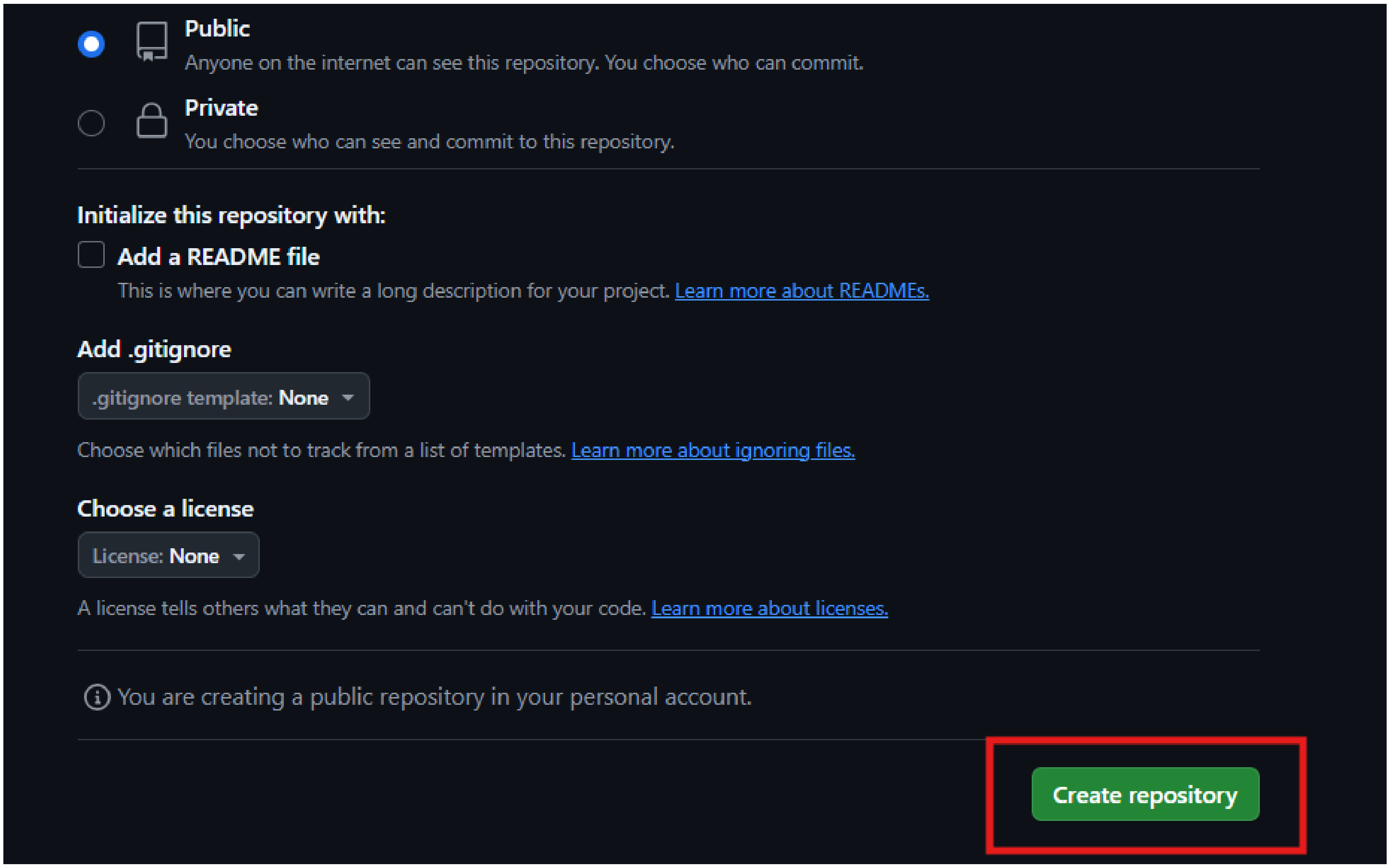
localhost:5173
Vite + React
count is 0
Edit src/App.jsx and save to test HMR
Click on the Vite and React logos to learn more

Initialize a Github Repository:

create a GitHub repository and commit the code

Repository name: *bucketlistapp*

Choose **Public**, then click **Create a new repository**



● Public
Anyone on the internet can see this repository. You choose who can commit.

○ Private
You choose who can see and commit to this repository.

Initialize this repository with:

□ Add a README file
This is where you can write a long description for your project. Learn more about READMEs.

Add .gitignore

.gitignore template: None

Choose which files not to track from a list of templates. Learn more about ignoring files.

Choose a license

License: None

A license tells others what they can and can't do with your code. Learn more about licenses.

You are creating a public repository in your personal account.

3. In VS Code terminal, navigating to the root folder of app (*bucketlistapp*) and run the following commands to initialize a Git repository and push the code:

```
git init

git add .

git commit -m "initial commit"

git remote add origin git@github.com:<your-username>/bucketlistapp.git

git branch -M main

git push -u origin main
```

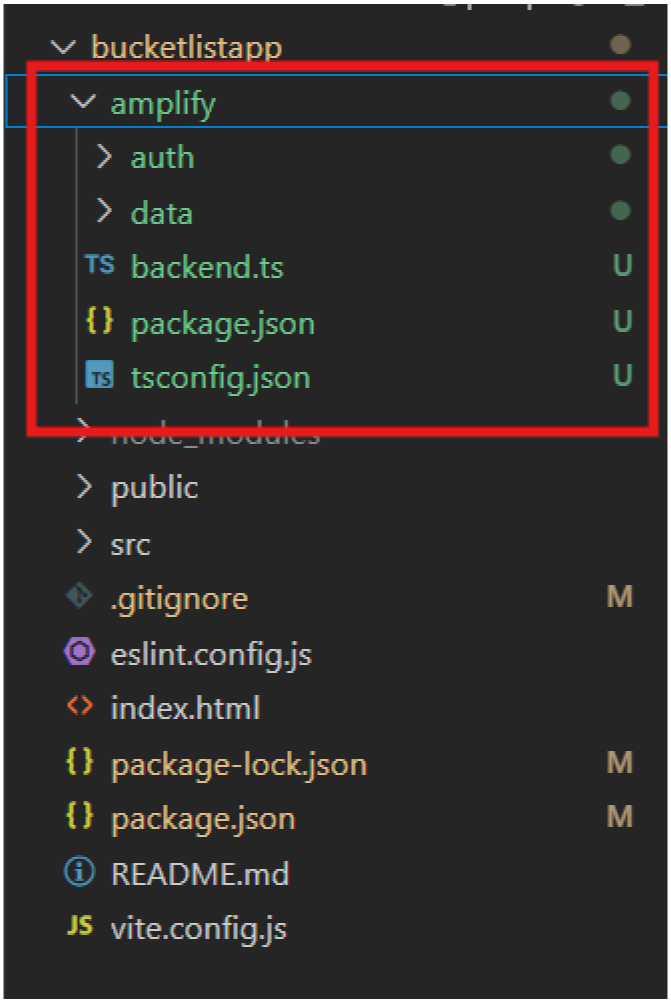
Install the Amplify Packages:

In this step, I'll scaffold an Amplify project for the app.

1. Run the following command in the terminal from app's root folder:

```
npm create amplify@latest -y
```

This will scaffold a lightweight Amplify project in the app directory.



```
∨ bucketlistapp
  ∨ amplify
    > auth
    > data
    TS backend.ts          U
    {} package.json        U
    TS tsconfig.json       U
  > node_modules
  > public
  > src
  .gitignore               M
  eslint.config.js
  index.html
  {} package-lock.json     M
  {} package.json          M
  README.md
  JS vite.config.js
```

2. Push the changes to GitHub using the following commands:

git add .

git commit -m "installing amplify"

git push origin main

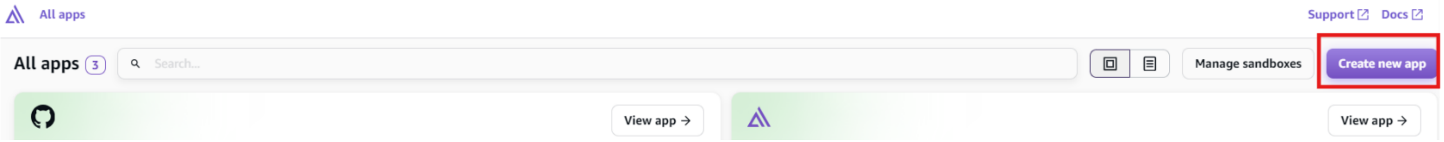
With this, i have successfully installed the Amplify packages.

Deploying App to AWS Amplify

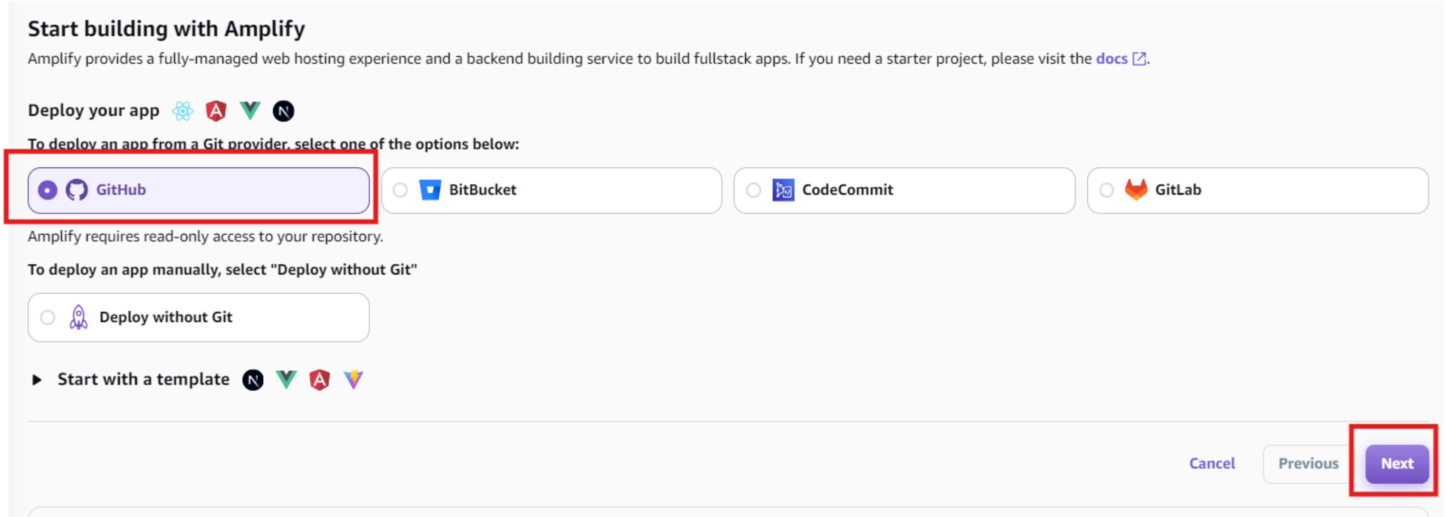
Now, i will connect GitHub repository to AWS Amplify, which will enable to build, deploy, and host my bucket list tracker app.

1. Sign in to the AWS Management Console and go to the AWS Amplify console at https://console.aws.amazon.com/amplify.

2. Click **Create new app.**



3. On the "Start building with Amplify" page, select **GitHub** for "Deploy your app" and click **Next**.



Start building with Amplify

Amplify provides a fully-managed web hosting experience and a backend building service to build fullstack apps. If you need a starter project, please visit the docs.

Deploy your app

To deploy an app from a Git provider, select one of the options below:

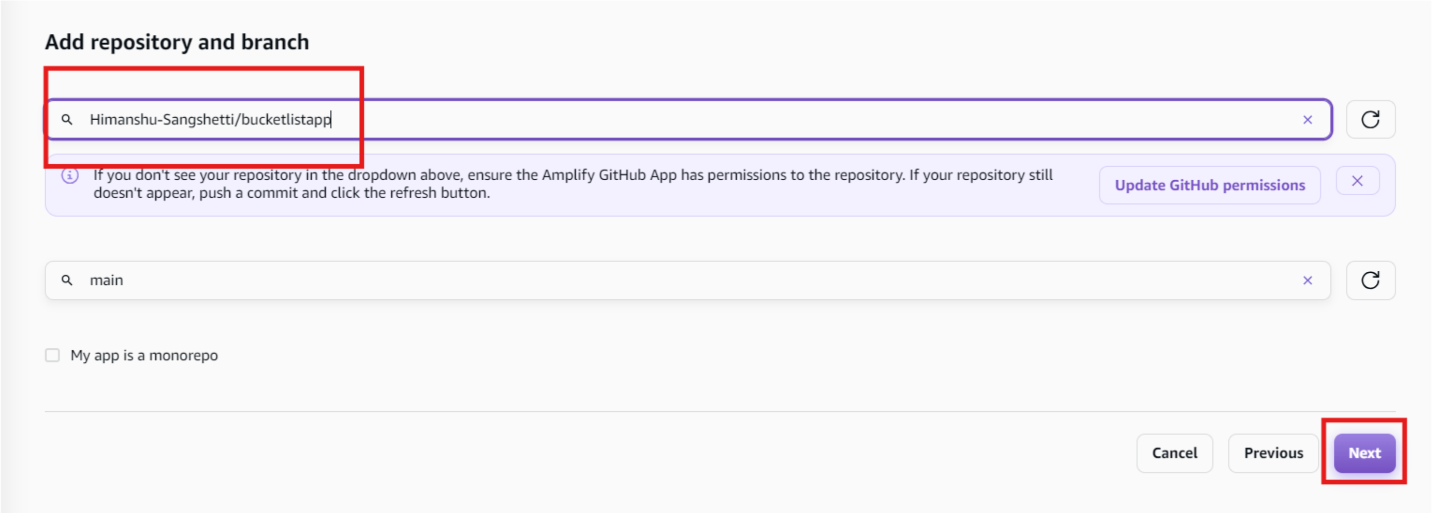
● GitHub ○ BitBucket ○ CodeCommit ○ GitLab

Amplify requires read-only access to your repository.

To deploy an app manually, select "Deploy without Git"

○ Deploy without Git

4. Authenticate with GitHub when prompted and select the repository and branch (main) created earlier. Click **Next**.



Add repository and branch

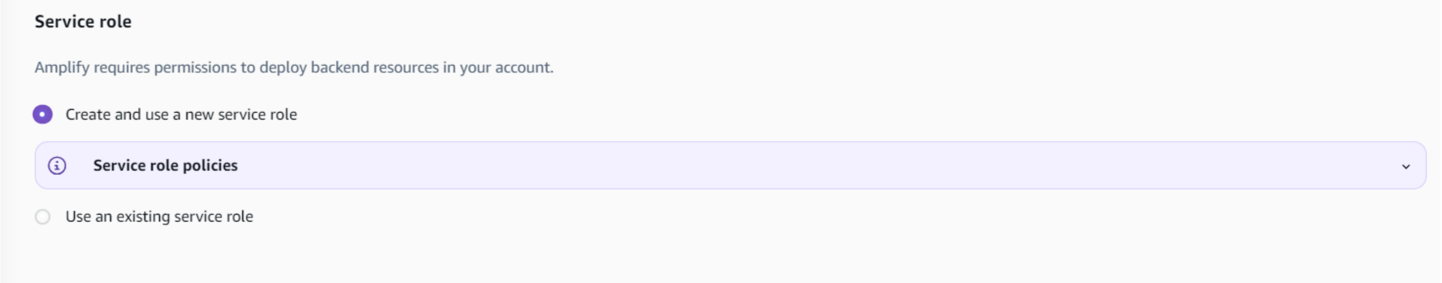
Himanshu-Sangshetti/bucketlistapp

If you don't see your repository in the dropdown above, ensure the Amplify GitHub App has permissions to the repository. If your repository still doesn't appear, push a commit and click the refresh button.

main

□ My app is a monorepo

5. In the service role section. Click on **Create and use a new service role**. Keep the rest as default and click **Next**.



Service role

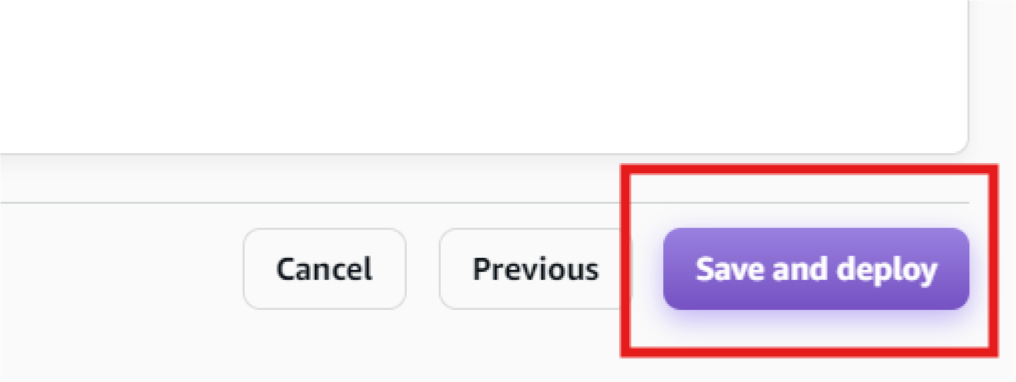
Amplify requires permissions to deploy backend resources in your account.

● Create and use a new service role

Service role policies

○ Use an existing service role

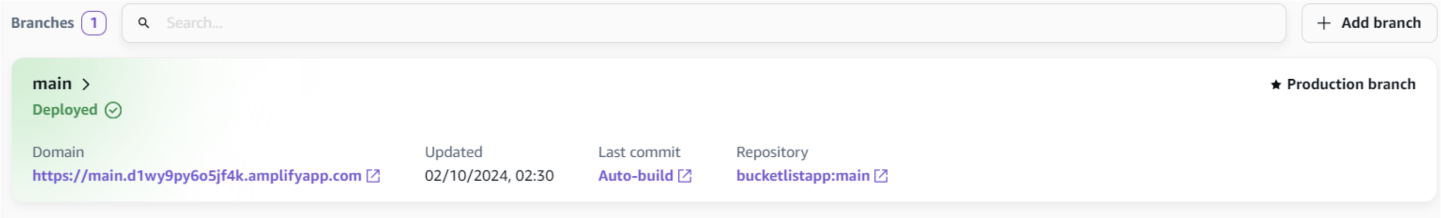
6. Reviewing selections and click **Save and deploy**.



Wait for AWS Amplify to create and deploy the app.

Creating a role

And here we go.

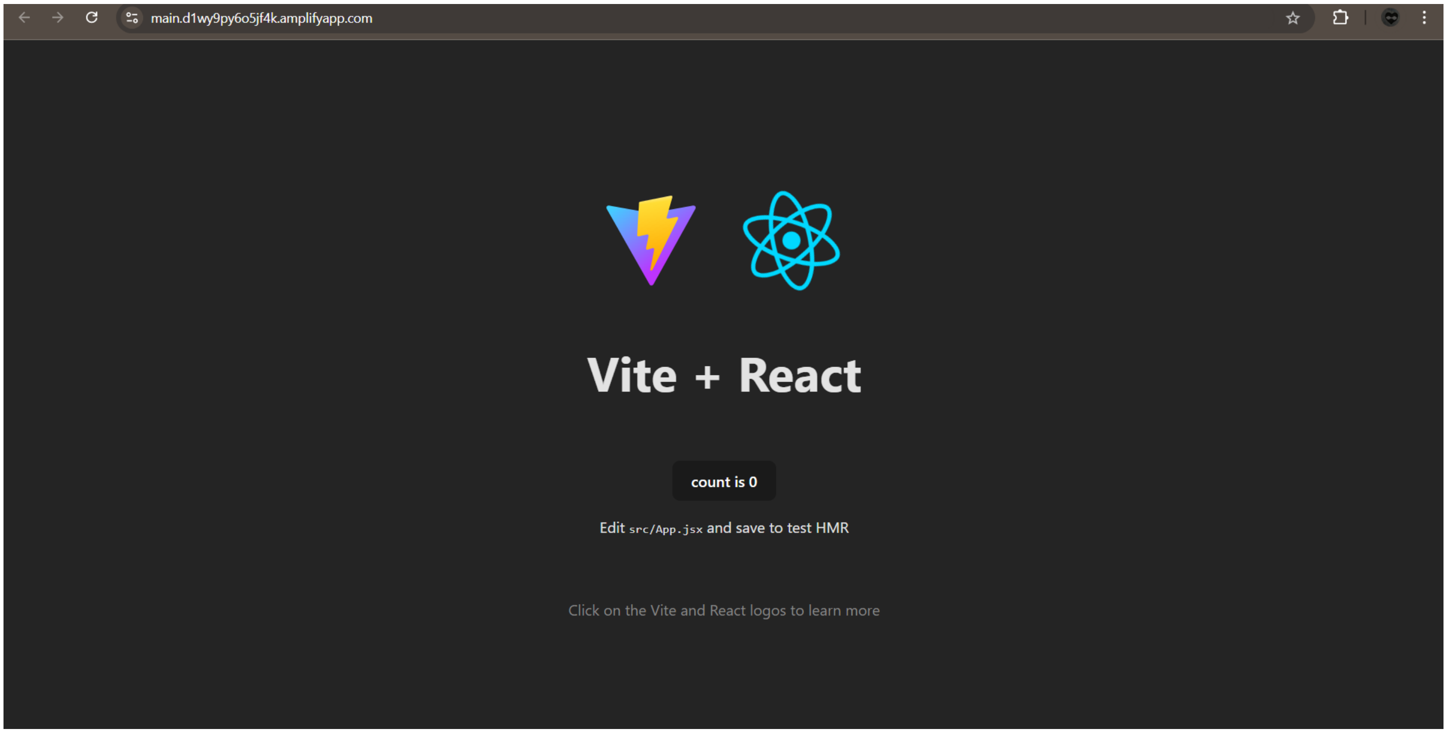


main >
Deployed

★ Production branch

Domain	Updated	Last commit	Repository
https://main.d1wy9py6o5jf4k.amplifyapp.com	02/10/2024, 02:30	Auto-build	bucketlistapp:main

check the website by clicking on the URL given in the Domain section.


main.d1wy9py6o5jf4k.amplifyapp.com

Vite + React

count is 0

Edit `src/App.jsx` and save to test HMR

Click on the Vite and React logos to learn more

AWS Amplify will now build source code and deploy app to a URL like https://...amplifyapp.com. Every time i push code to GitHub, Amplify will trigger a new deployment.

Setup Amplify Backend: Introduction

In this task, I will use AWS Amplify to provision a cloud backend for bucket list tracker application. This includes setting up authentication, data storage, and file storage, which will allow users to manage their bucket lists.

Now that I have a React-based bucket list tracker app, the next step is to configure a cloud backend using AWS Amplify.

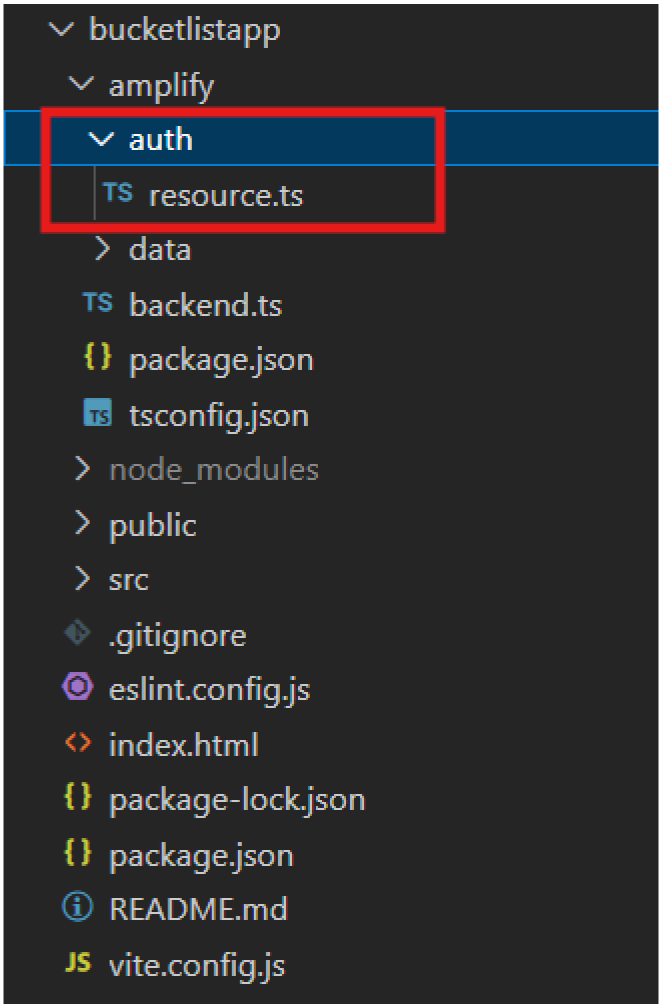
With Amplify, i can set up authentication, data storage, and file storage with unified developer experience.

Amplify allows to build and deploy full stack applications using a code-first approach, with built-in hosting, backend, and UI-building capabilities.

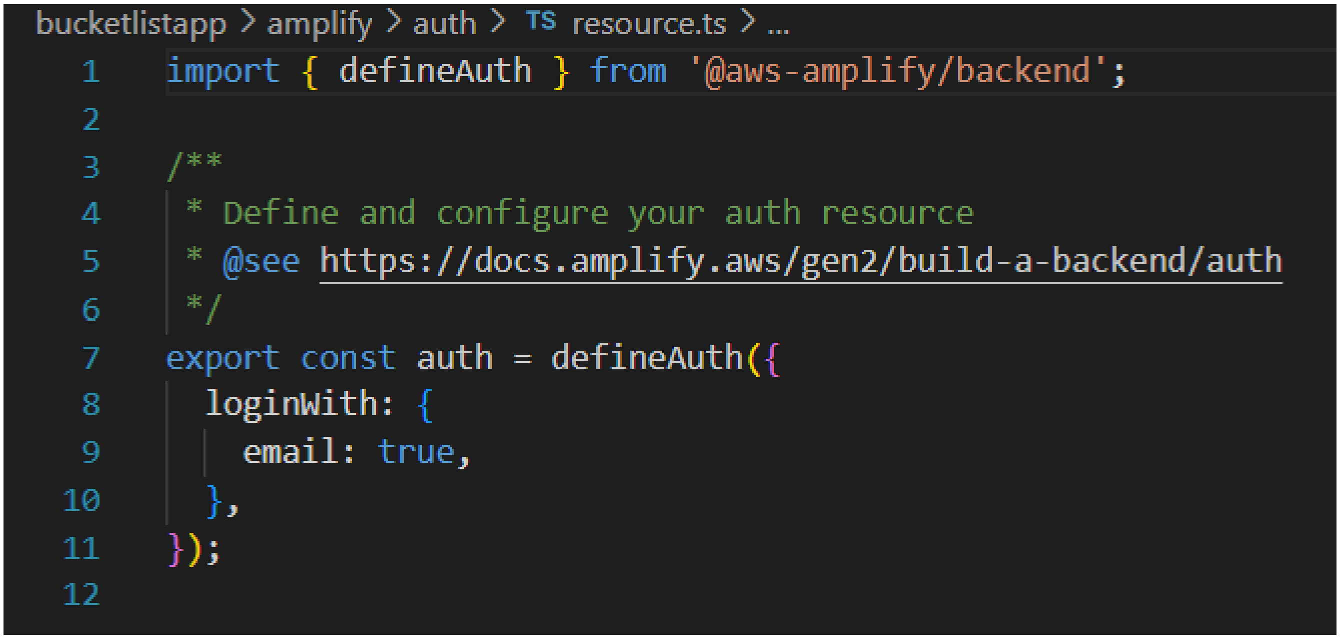
Setup Amplify Authentication

The app will allow users to sign up using email, and upon sign-up, will receive a verification email.

By default, authentication resource is configured in the bucketlistapp/amplify/auth/resource.ts file. Keeping the default authentication settings as they are.



```
bucketlistapp
  amplify
    auth
      resource.ts
    data
    backend.ts
    package.json
    tsconfig.json
  node_modules
  public
  src
  .gitignore
  eslint.config.js
  index.html
  package-lock.json
  package.json
  README.md
  vite.config.js
```



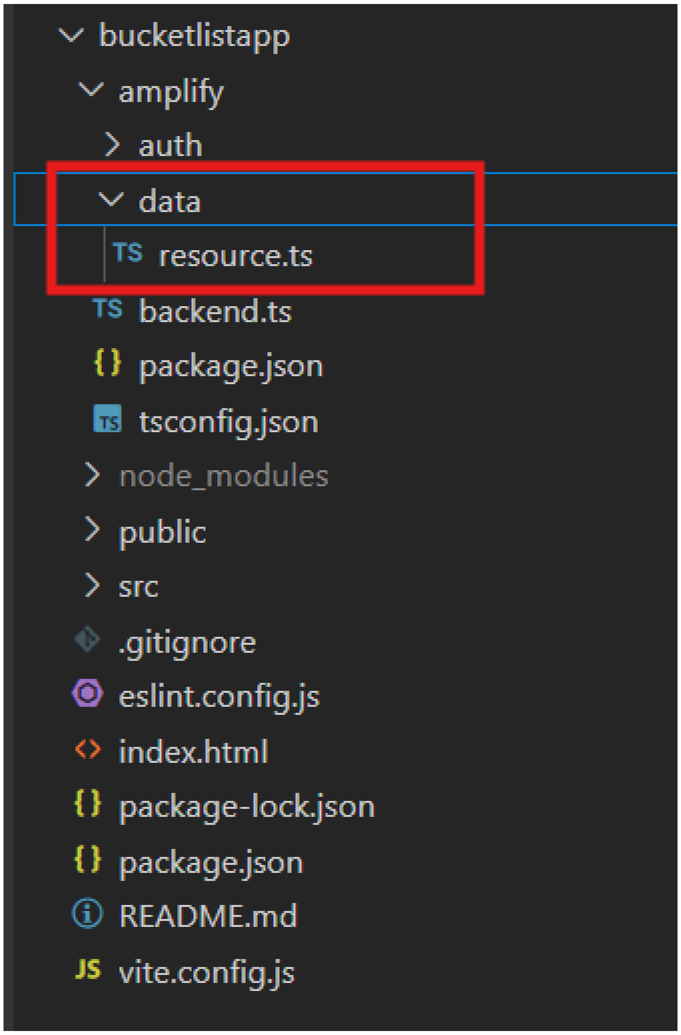
bucketlistapp > amplify > auth > resource.ts > ...

```
import { defineAuth } from '@aws-amplify/backend';

/**
 * Define and configure your auth resource
 * @see https://docs.amplify.aws/gen2/build-a-backend/auth
 */
export const auth = defineAuth({
  loginWith: {
    email: true,
  },
});

```

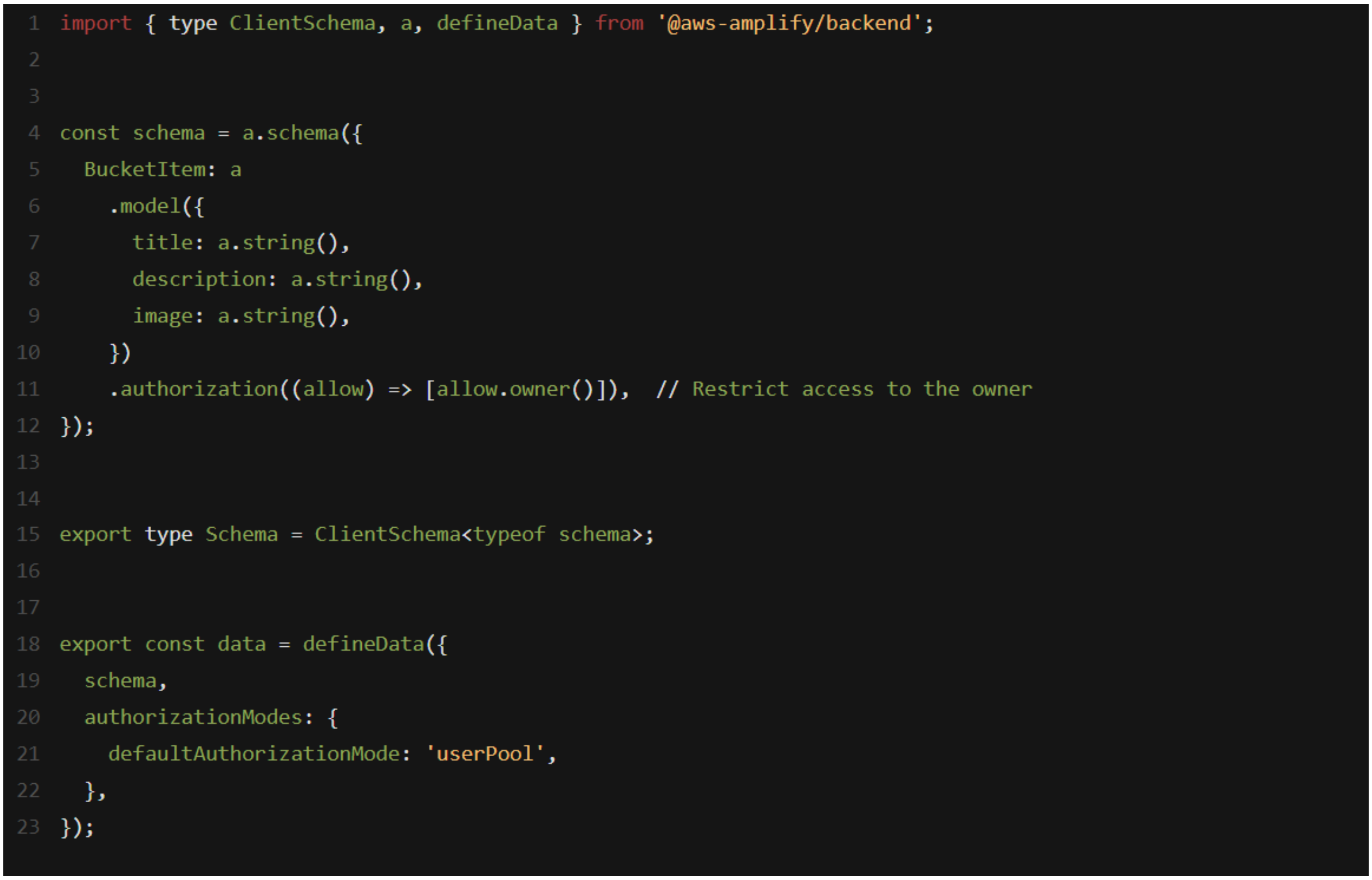
Setup Amplify Data

In this step, i will define a scheme for storing the bucket list items. Each user will be able to create, delete, and list their own bucket list entries, and Amplify will automatically handle per-user ownership of the data.

1.Navigating to the bucketlistapp/amplify/data/resource.ts file and updating it with the following code given below.



2. This code will define a model for the bucket list items, ensuring that only the owner can access their data.



```
import { type ClientSchema, a, defineData } from '@aws-amplify/backend';


const schema = a.schema({
  BucketItem: a
    .model({
      title: a.string(),
      description: a.string(),
      image: a.string(),
    })
    .authorization((allow) => [allow.owner()]),  // Restrict access to the owner
});


export type Schema = ClientSchema<typeof schema>;


export const data = defineData({
  schema,
  authorizationModes: {
    defaultAuthorizationMode: 'userPool',
  },
});
```

In this schema:

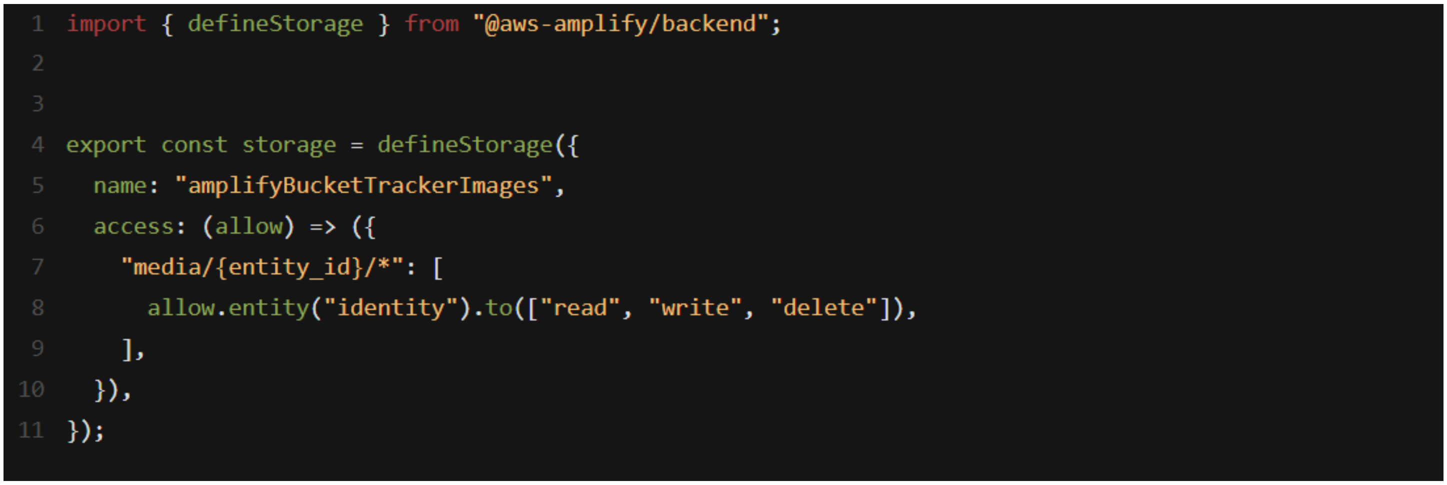
- Each bucket list item includes a title, description, and a completed status.
- The authorization rule ensures that only the user who created the item can access it.

Setup Amplify Storage:

I will set up file storage to allow users to upload images related to their bucket list items (optional feature).

1. Creating a new folder called storage inside the bucketlistapp/amplify folder, and inside that, creating a new file named resource.ts .

2. Updating the resource.ts file with the following code to configure storage for images:



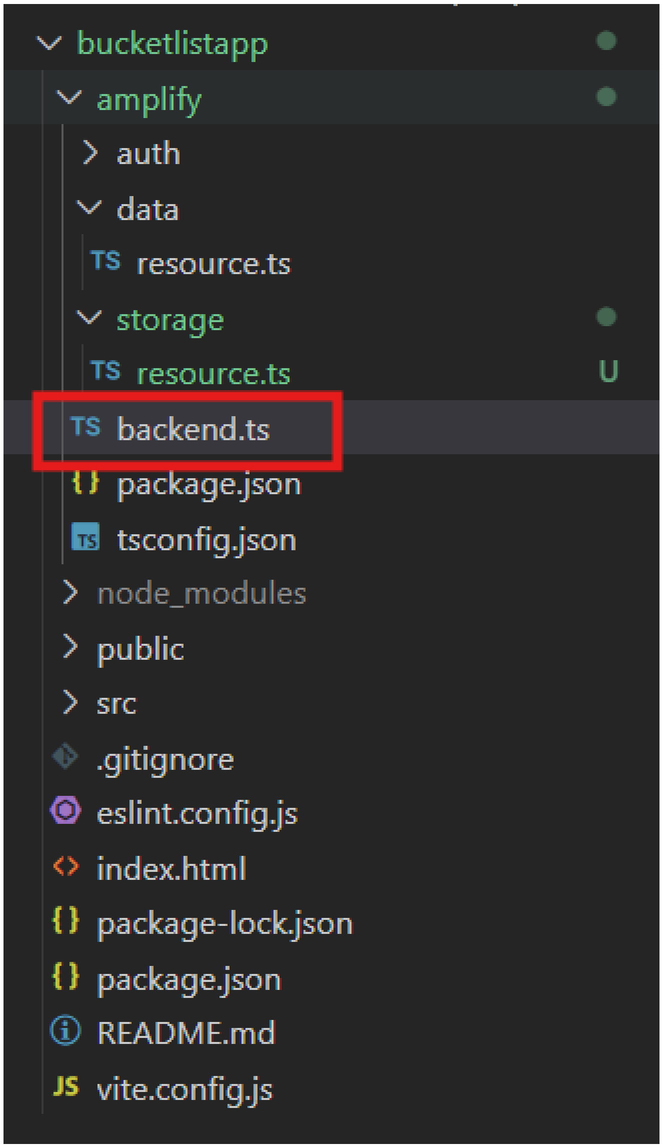
```
import { defineStorage } from "@aws-amplify/backend";


export const storage = defineStorage({
  name: "amplifyBucketTrackerImages",
  access: (allow) => ({
    "media/{entity_id}/*": [
      allow.entity("identity").to(["read", "write", "delete"]),
    ],
  }),
});
```

This storage configuration ensures that only the person who uploads the image can access it. The entity_id will be replaced with the user's identifier during file uploads, restricting access to the file.

Finalize Backend Setup

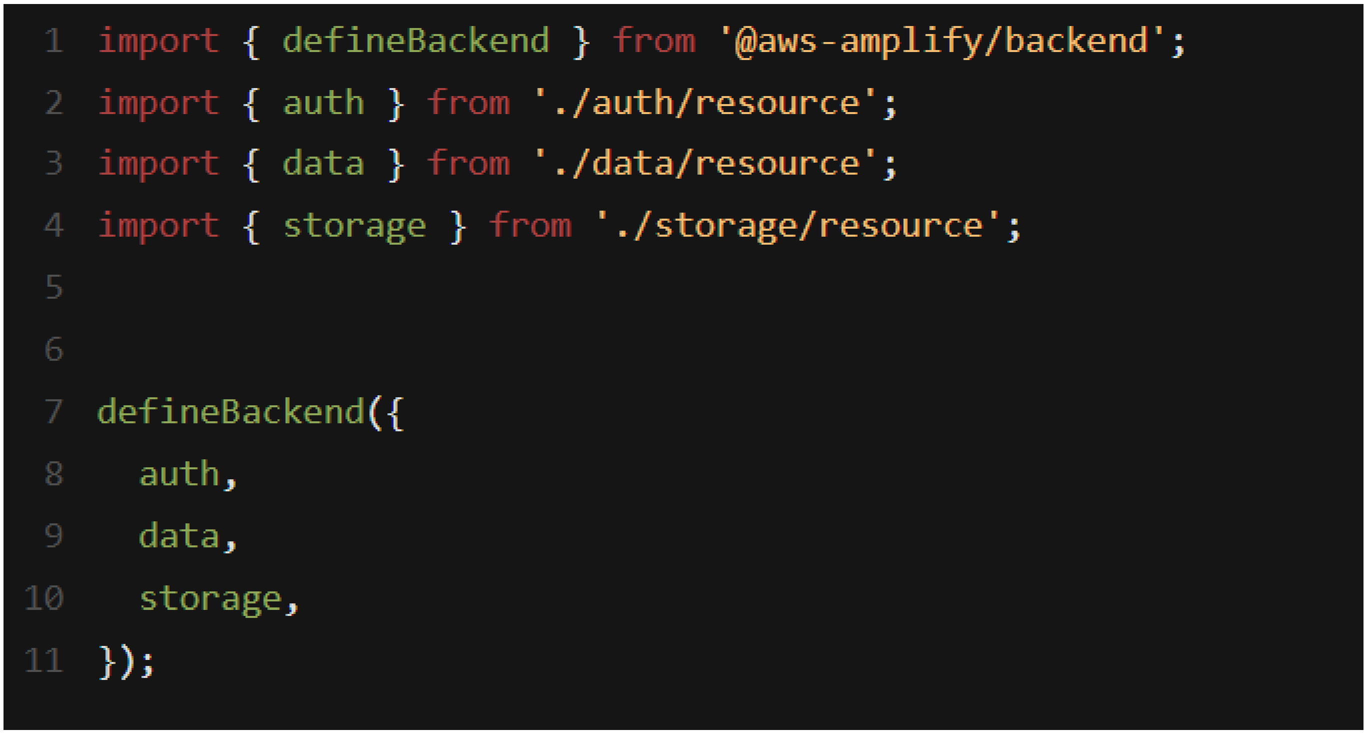
Now, i need to link the authentication, data, and storage resources in my Amplify backend configuration.

navigating to the amplify/backend.ts file and update it with the following code given below:

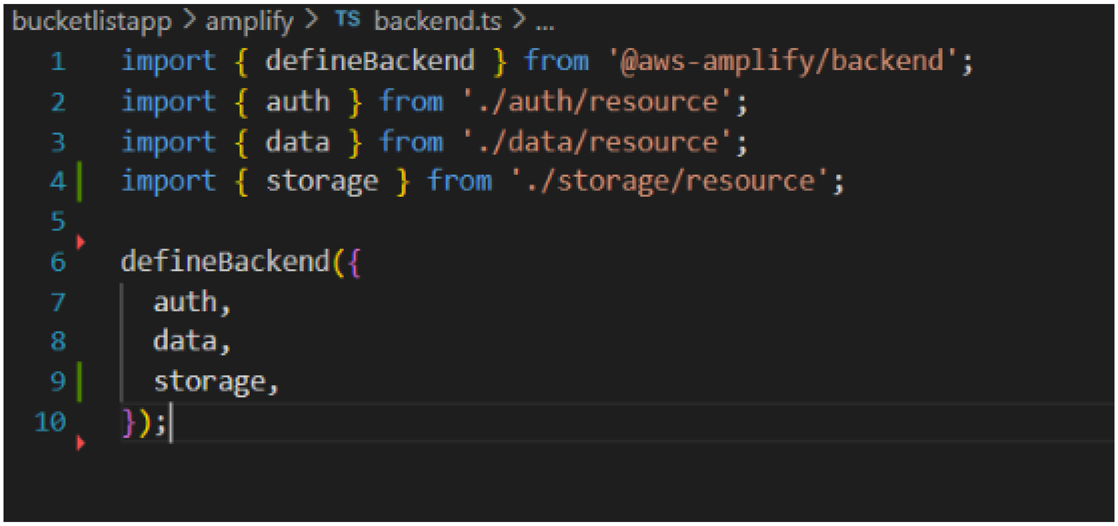


```
bucketlistapp
  amplify
    auth
    data
      resource.ts
    storage
      resource.ts        U
    backend.ts
    package.json
    tsconfig.json
  node_modules
  public
  src
  .gitignore
  eslint.config.js
  index.html
  package-lock.json
  package.json
  README.md
  vite.config.js
```

The code:



```
import { defineBackend } from '@aws-amplify/backend';
import { auth } from './auth/resource';
import { data } from './data/resource';
import { storage } from './storage/resource';


defineBackend({
  auth,
  data,
  storage,
});
```



```
import { defineBackend } from '@aws-amplify/backend';
import { auth } from './auth/resource';
import { data } from './data/resource';
import { storage } from './storage/resource';

defineBackend({
  auth,
  data,
  storage,
});
```

This ensures that all the backend resources (auth, data, and storage) are properly configured and linked.

Deploy the Amplify Backend in a Cloud Sandbox

Now, I'll deploy Amplify backend resources to a personal cloud sandbox environment, which allows to rapidly build, test, and iterate on the app.

1. In a new terminal window, running the following command from app's root folder:

npx ampx sandbox

After running the above command, get the following error: *SSMCredentialsError: UnrecognizedClientException: The security token included in the request is invalid.*

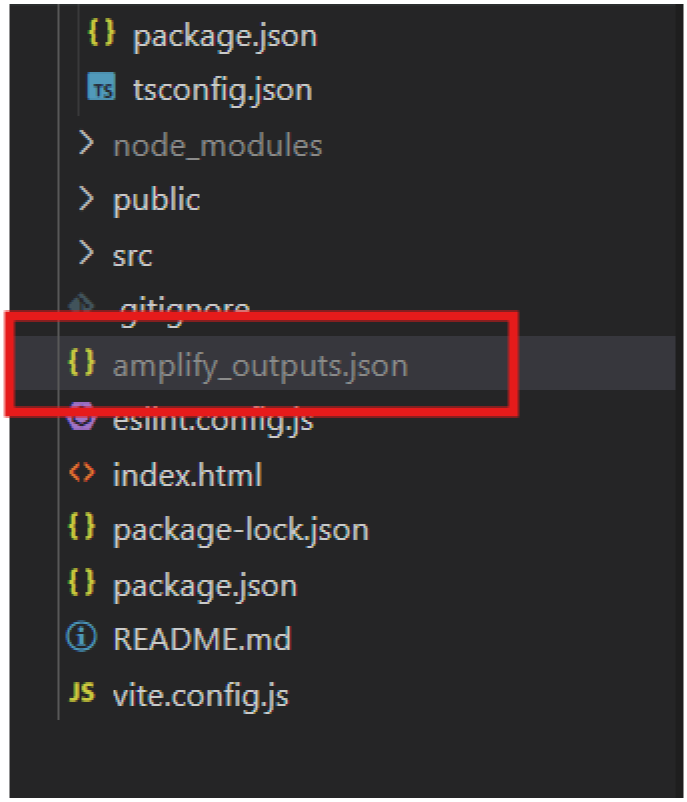
Run this command: aws configure

I can get the access key and secret access key by navigating to AWS Console > IAM > Users

If User is available, navigate to the security credentials to get the keys. If no user is available, create one by clicking on Create User.

This command starts a Cloud sandbox, which is an isolated development environment connected to AWS Cloud resources. Each developer can use their own disposable sandbox for rapid iteration.

2. Once the sandbox has been fully deployed, will see a confirmation message, and an amplify_outputs.json file will be generated and added to the project.



With these steps, bucket list tracker application now has a fully configured backend, including user authentication, data storage, and file storage.

ready to move forward with implementing the application's frontend logic to interact with the backend services.

Connecting Frontend and Backend:

In this task, I will build the front end of bucket list tracker app and connect it to the cloud backend I have already set up.

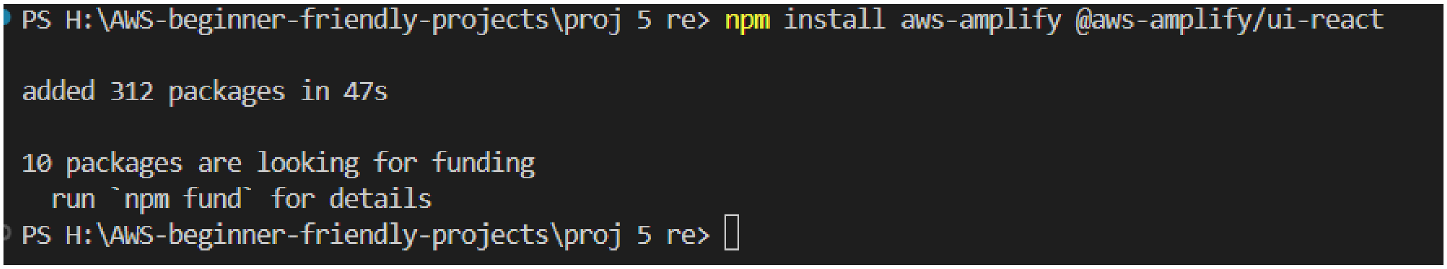
I will use AWS Amplify's UI component library to create complete user authentication flow and implement the ability to create, update, and delete bucket list items.

Additionally, i will create the front end of the bucket list tracker, where users can add, update, and delete items on their bucket list. They will also be able to upload images associated with each item.

Install Amplify Libraries

In a new terminal window, navigate to the project folder (bucketlistapp), and running the following command to install the necessary Amplify libraries:

npm install aws-amplify @aws-amplify/ui-react

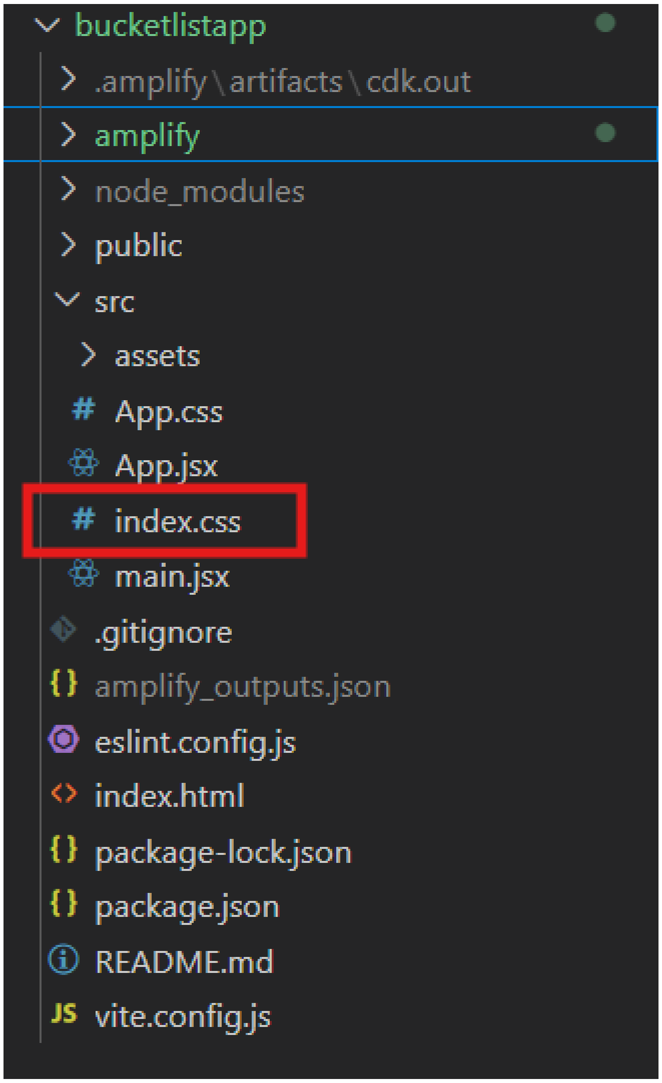


```
PS H:\AWS-beginner-friendly-projects\proj 5 re> npm install aws-amplify @aws-amplify/ui-react

added 312 packages in 47s

10 packages are looking for funding
  run `npm fund` for details
PS H:\AWS-beginner-friendly-projects\proj 5 re> 
```

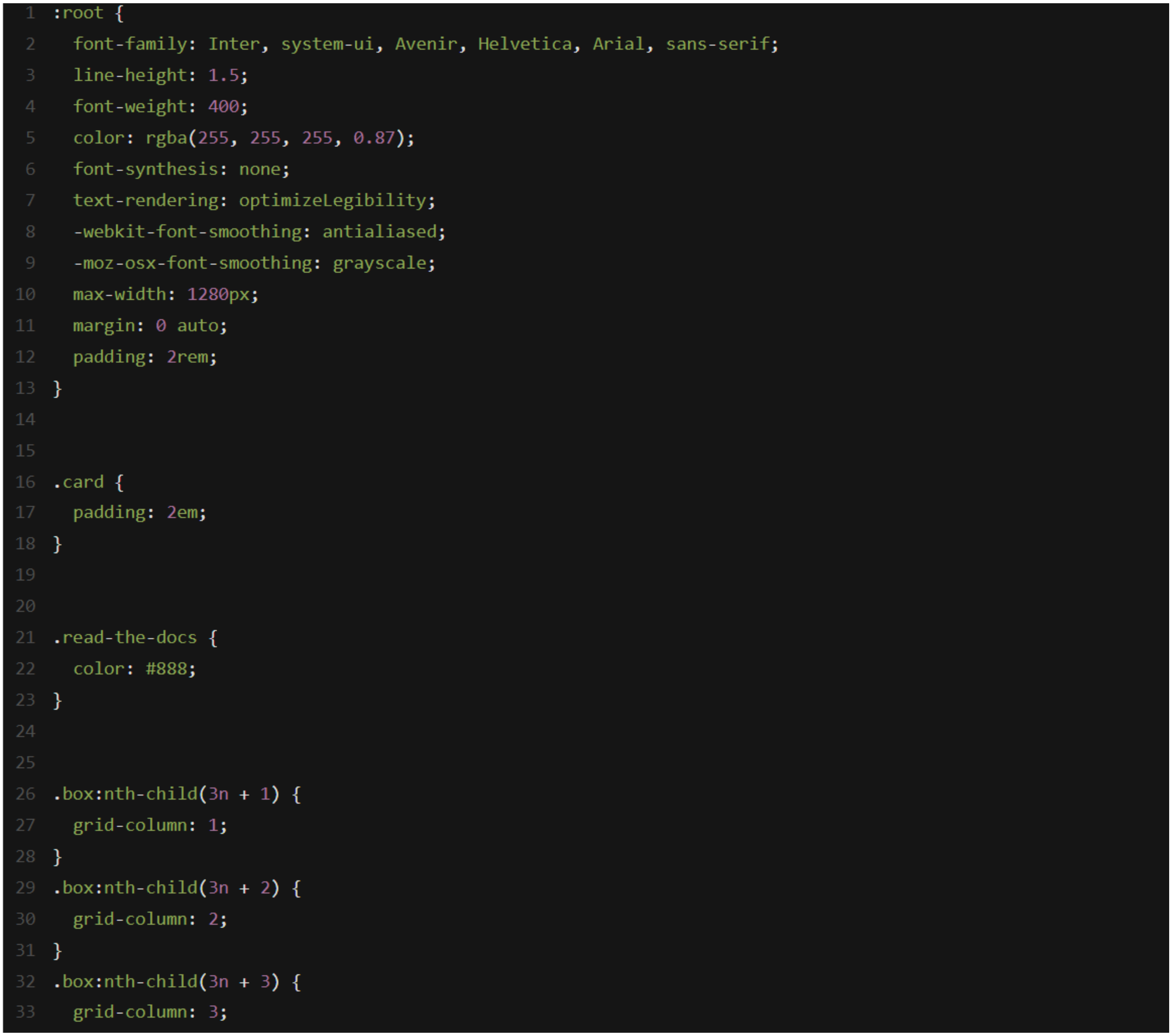
These libraries include the client-side APIs to connect app's frontend to the backend services and the UI components for authentication.

UI Setup and Styling

1. navigating to the bucketlistapp/src/index.css file.



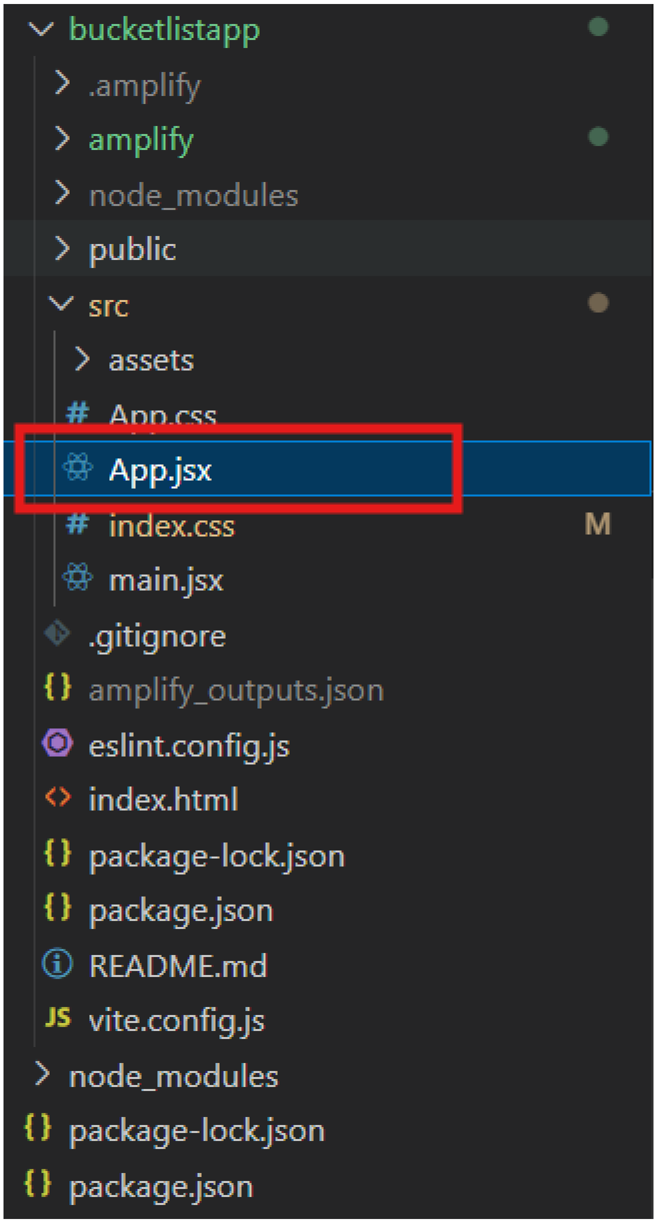
2. Update it with the following code to style the bucket list UI:



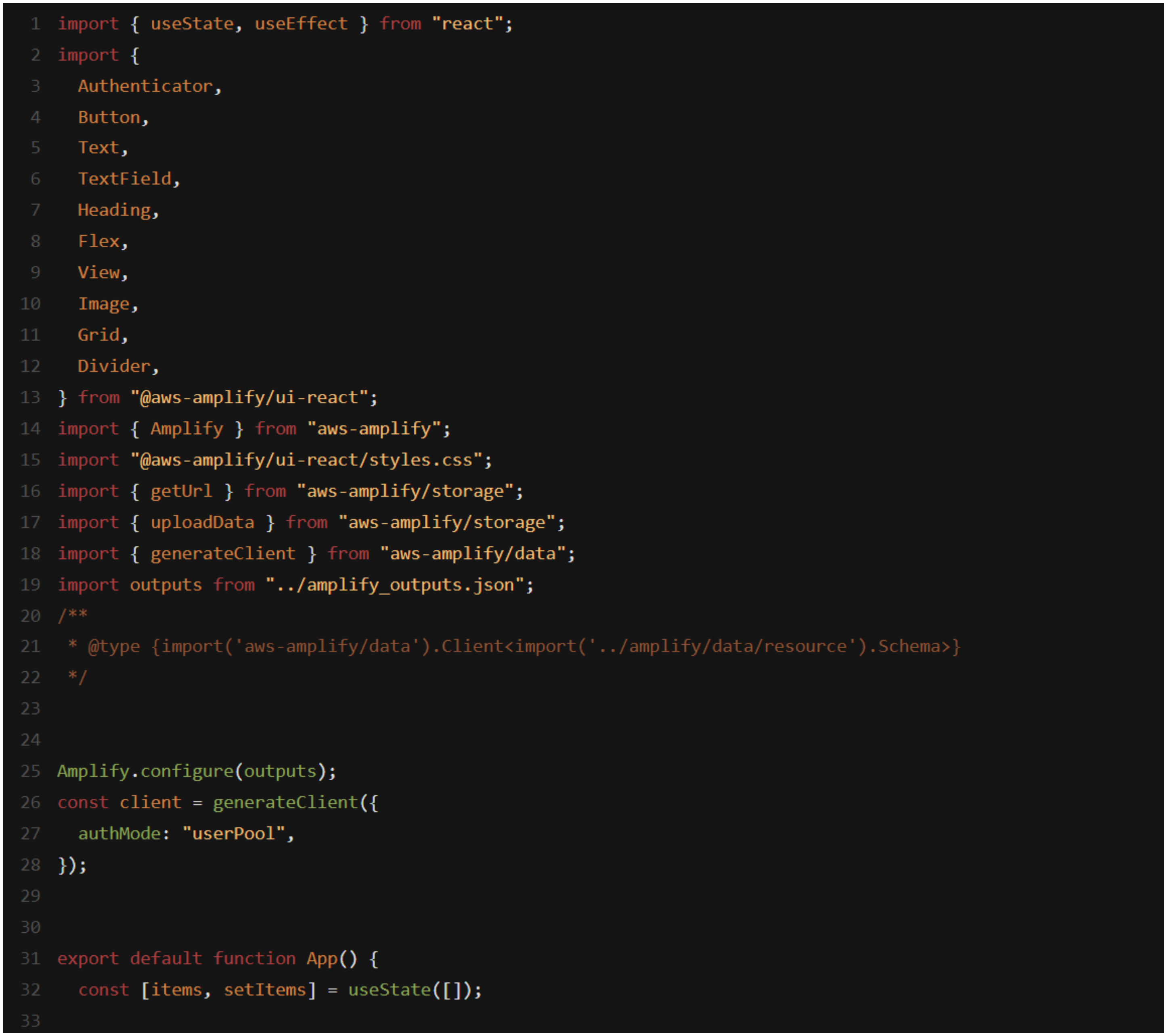
```css
:root {
  font-family: Inter, system-ui, Avenir, Helvetica, Arial, sans-serif;
  line-height: 1.5;
  font-weight: 400;
  color: rgba(255, 255, 255, 0.87);
  font-synthesis: none;
  text-rendering: optimizeLegibility;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
  max-width: 1280px;
  margin: 0 auto;
  padding: 2rem;
}


.card {
  padding: 2em;
}


.read-the-docs {
  color: #888;
}


.box:nth-child(3n + 1) {
  grid-column: 1;
}
.box:nth-child(3n + 2) {
  grid-column: 2;
}
.box:nth-child(3n + 3) {
  grid-column: 3;
```

This will set the layout and styles for the bucket list UI, making it responsive and visually appealing.

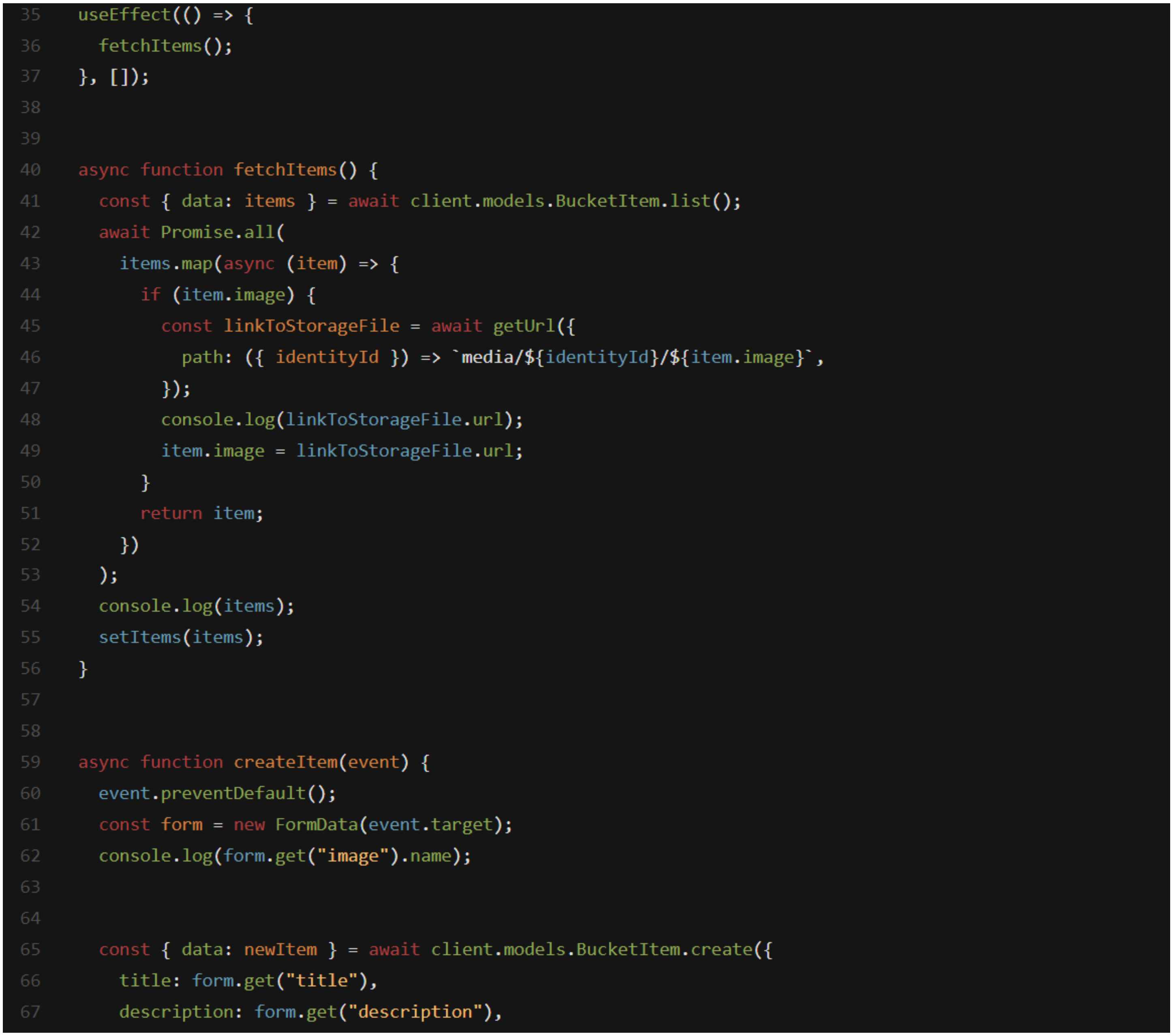
3.navigating to the bucketlistapp/src/App.jsx file.



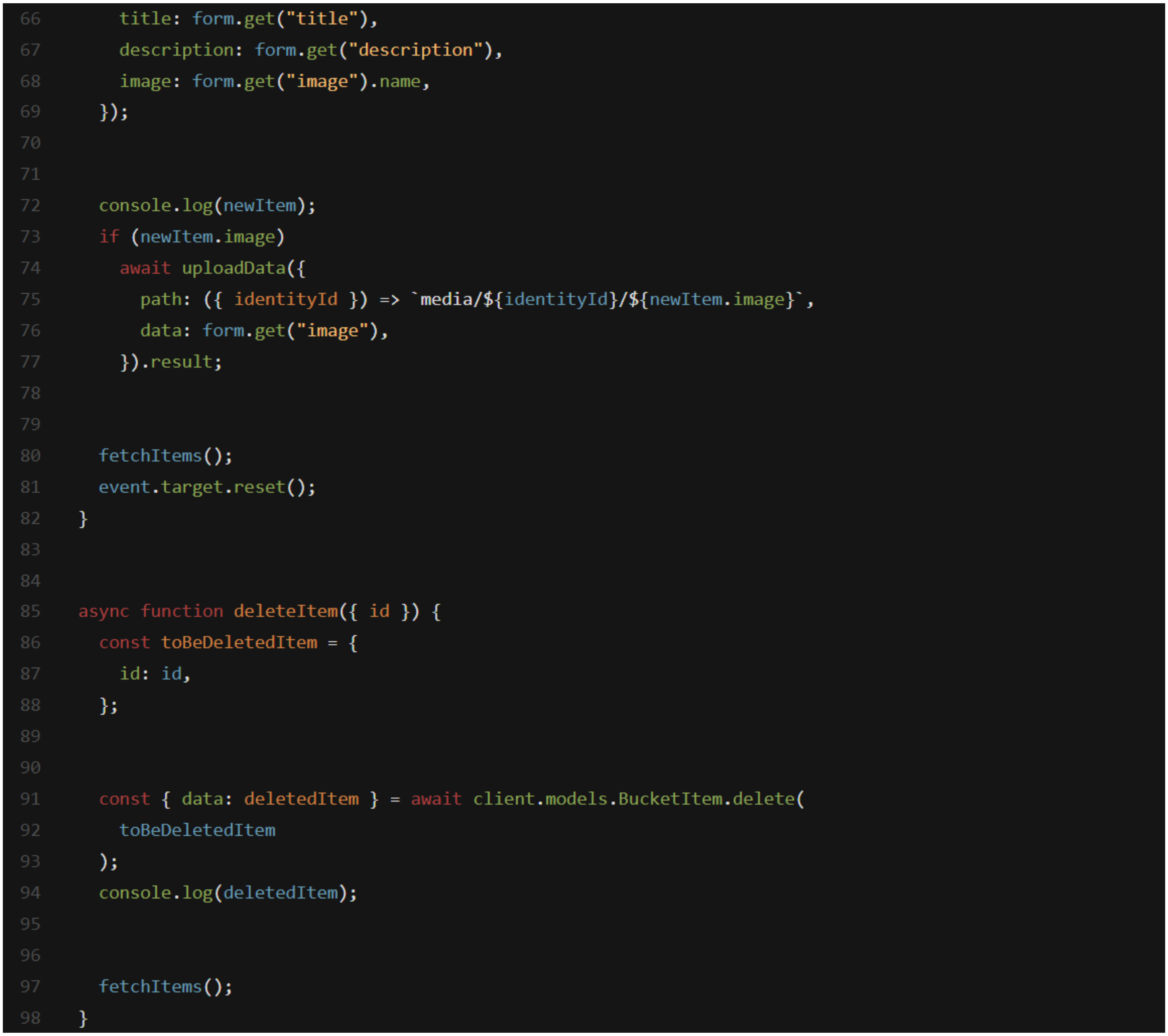
4. Update it with the following code to to App.jsx:



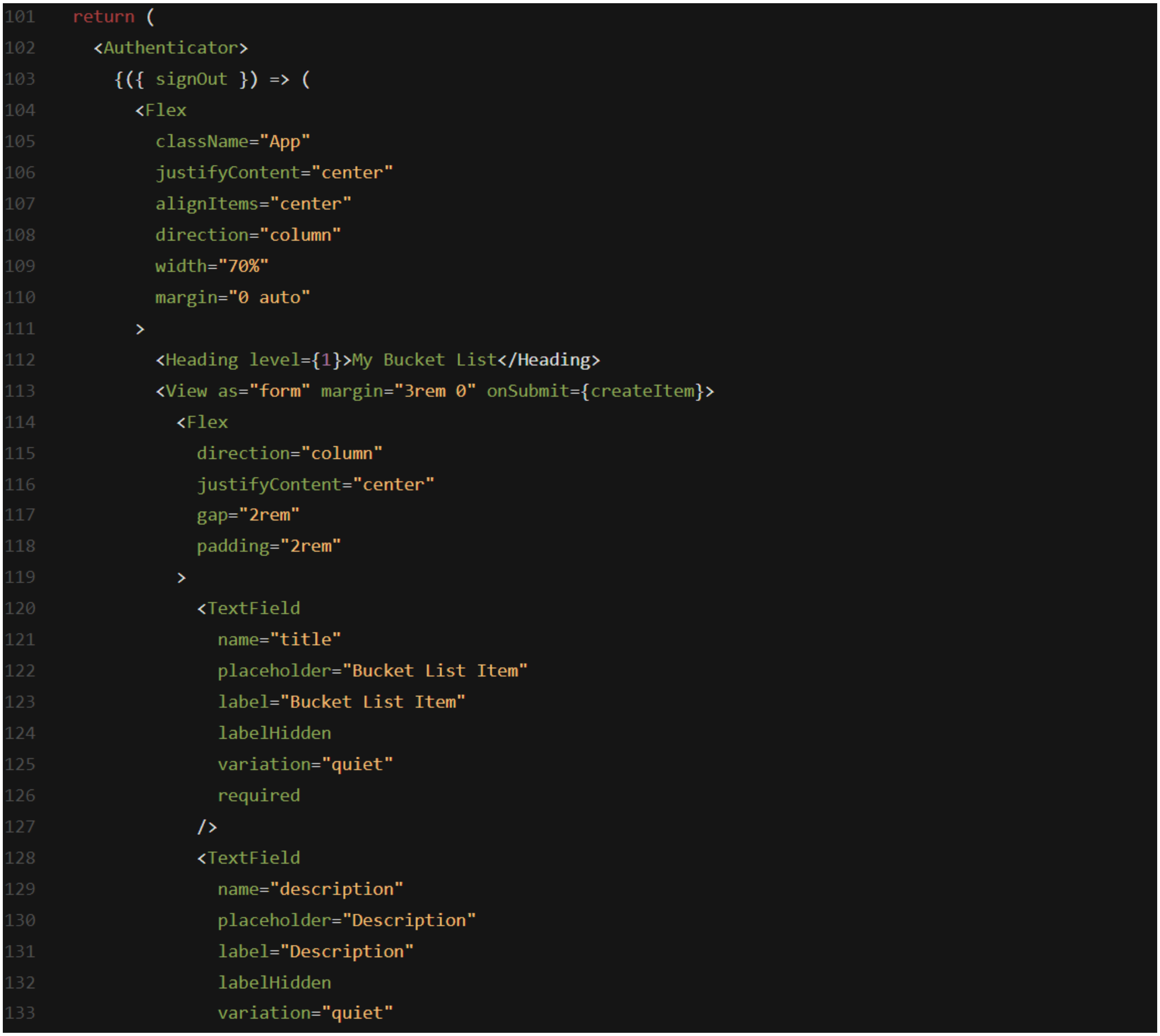
```
import { useState, useEffect } from "react";
import {
  Authenticator,
  Button,
  Text,
  TextField,
  Heading,
  Flex,
  View,
  Image,
  Grid,
  Divider,
} from "@aws-amplify/ui-react";
import { Amplify } from "aws-amplify";
import "@aws-amplify/ui-react/styles.css";
import { getUrl } from "aws-amplify/storage";
import { uploadData } from "aws-amplify/storage";
import { generateClient } from "aws-amplify/data";
import outputs from "../amplify_outputs.json";
/**
 * @type {import('aws-amplify/data').Client<import('../amplify/data/resource').Schema>}
 */


Amplify.configure(outputs);
const client = generateClient({
  authMode: "userPool",
});


export default function App() {
  const [items, setItems] = useState([]);

```



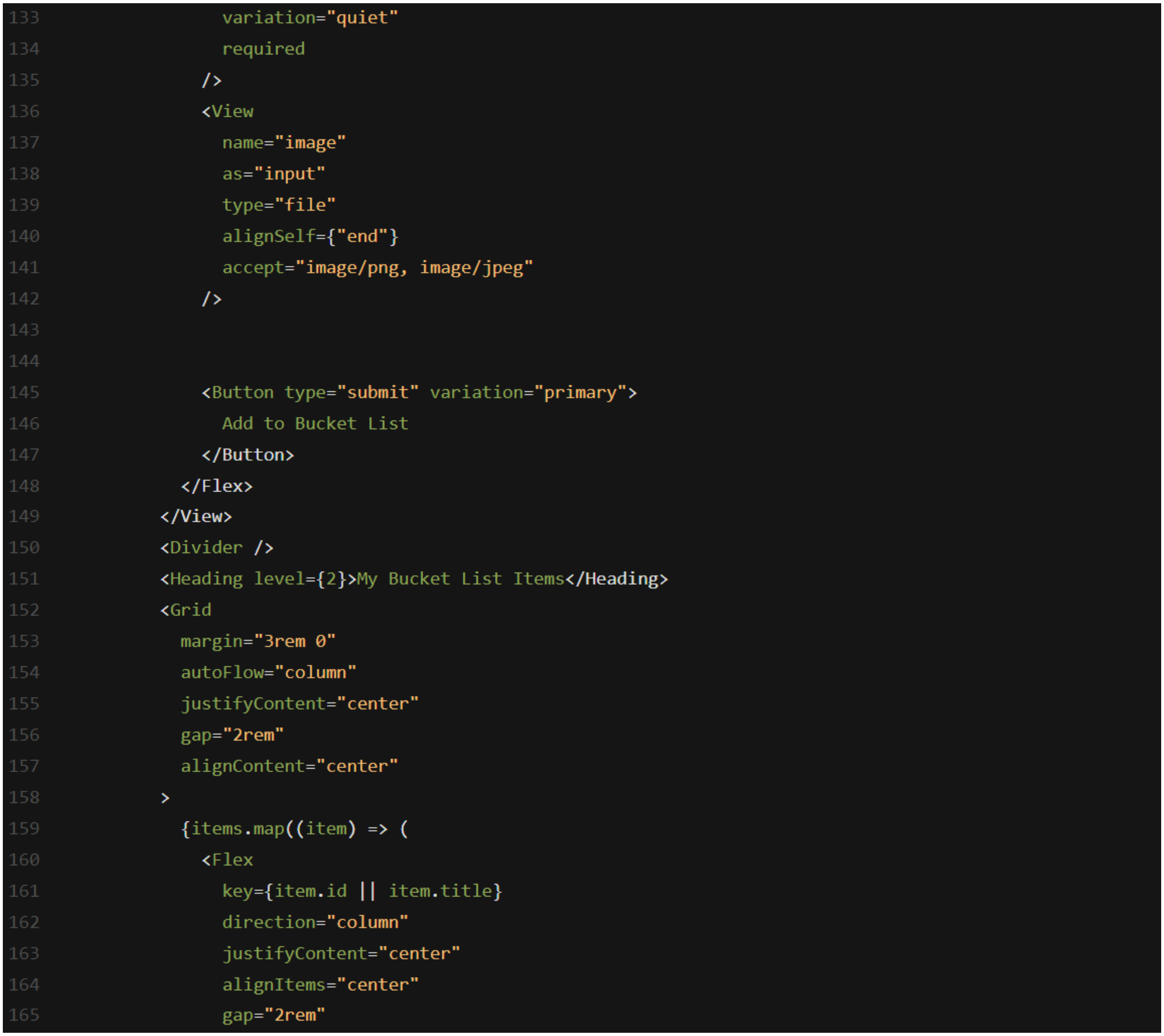
```
useEffect(() => {
  fetchItems();
}, []);


async function fetchItems() {
  const { data: items } = await client.models.BucketItem.list();
  await Promise.all(
    items.map(async (item) => {
      if (item.image) {
        const linkToStorageFile = await getUrl({
          path: ({ identityId }) => `media/${identityId}/${item.image}`,
        });
        console.log(linkToStorageFile.url);
        item.image = linkToStorageFile.url;
      }
      return item;
    })
  );
  console.log(items);
  setItems(items);
}


async function createItem(event) {
  event.preventDefault();
  const form = new FormData(event.target);
  console.log(form.get("image").name);


  const { data: newItem } = await client.models.BucketItem.create({
    title: form.get("title"),
    description: form.get("description"),
```



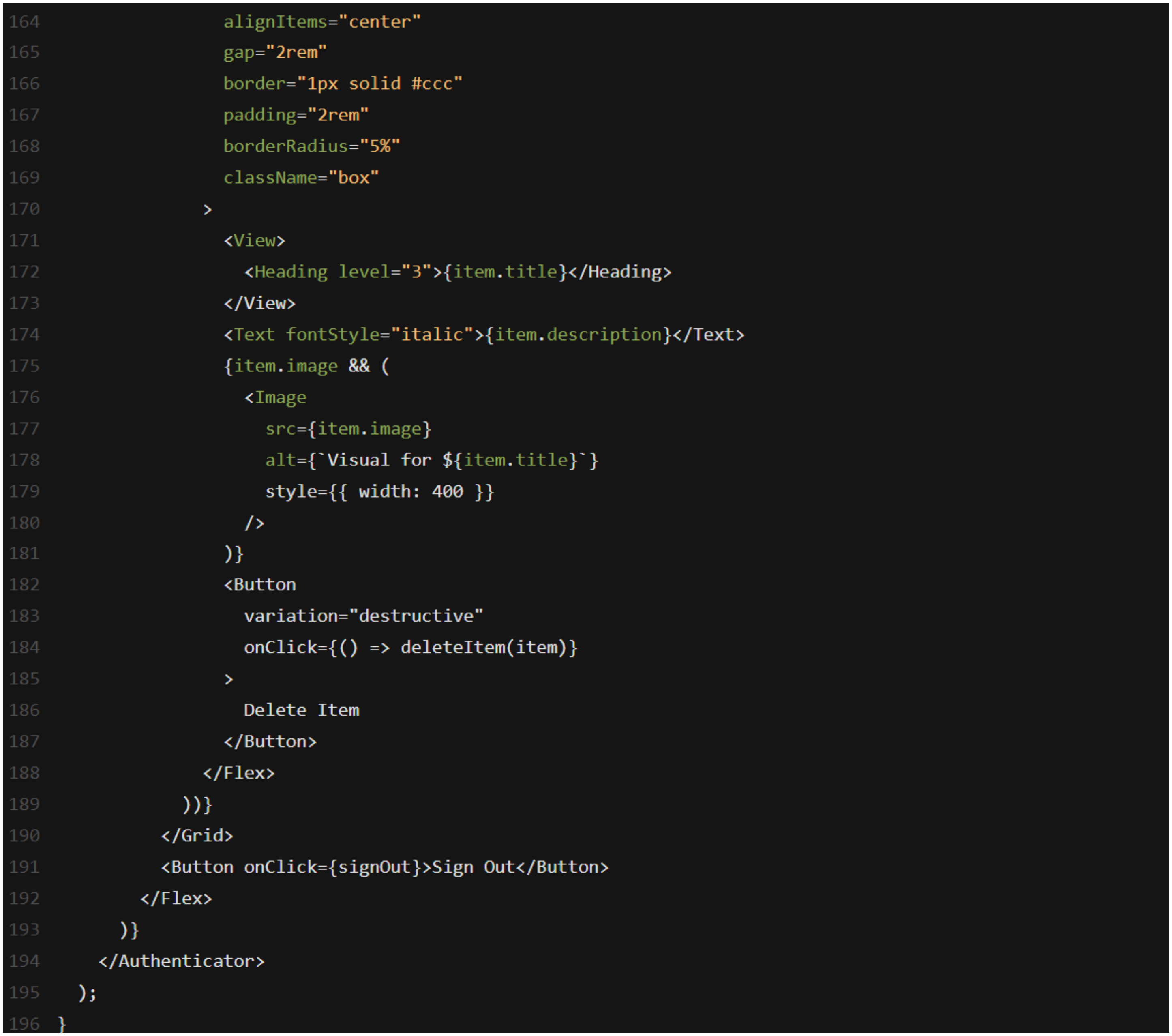
```
    title: form.get("title"),
    description: form.get("description"),
    image: form.get("image").name,
  });


  console.log(newItem);
  if (newItem.image)
    await uploadData({
      path: ({ identityId }) => `media/${identityId}/${newItem.image}`,
      data: form.get("image"),
    }).result;


  fetchItems();
  event.target.reset();
}


async function deleteItem({ id }) {
  const toBeDeletedItem = {
    id: id,
  };


  const { data: deletedItem } = await client.models.BucketItem.delete(
    toBeDeletedItem
  );
  console.log(deletedItem);


  fetchItems();
}
```



```
  return (
    <Authenticator>
      {({ signOut }) => (
        <Flex
          className="App"
          justifyContent="center"
          alignItems="center"
          direction="column"
          width="70%"
          margin="0 auto"
        >
          <Heading level={1}>My Bucket List</Heading>
          <View as="form" margin="3rem 0" onSubmit={createItem}>
            <Flex
              direction="column"
              justifyContent="center"
              gap="2rem"
              padding="2rem"
            >
              <TextField
                name="title"
                placeholder="Bucket List Item"
                label="Bucket List Item"
                labelHidden
                variation="quiet"
                required
              />
              <TextField
                name="description"
                placeholder="Description"
                label="Description"
                labelHidden
                variation="quiet"
```



```
              variation="quiet"
              required
            />
            <View
              name="image"
              as="input"
              type="file"
              alignSelf={"end"}
              accept="image/png, image/jpeg"
            />


            <Button type="submit" variation="primary">
              Add to Bucket List
            </Button>
          </Flex>
        </View>
        <Divider />
        <Heading level={2}>My Bucket List Items</Heading>
        <Grid
          margin="3rem 0"
          autoFlow="column"
          justifyContent="center"
          gap="2rem"
          alignContent="center"
        >
          {items.map((item) => (
            <Flex
              key={item.id || item.title}
              direction="column"
              justifyContent="center"
              alignItems="center"
              gap="2rem"
```



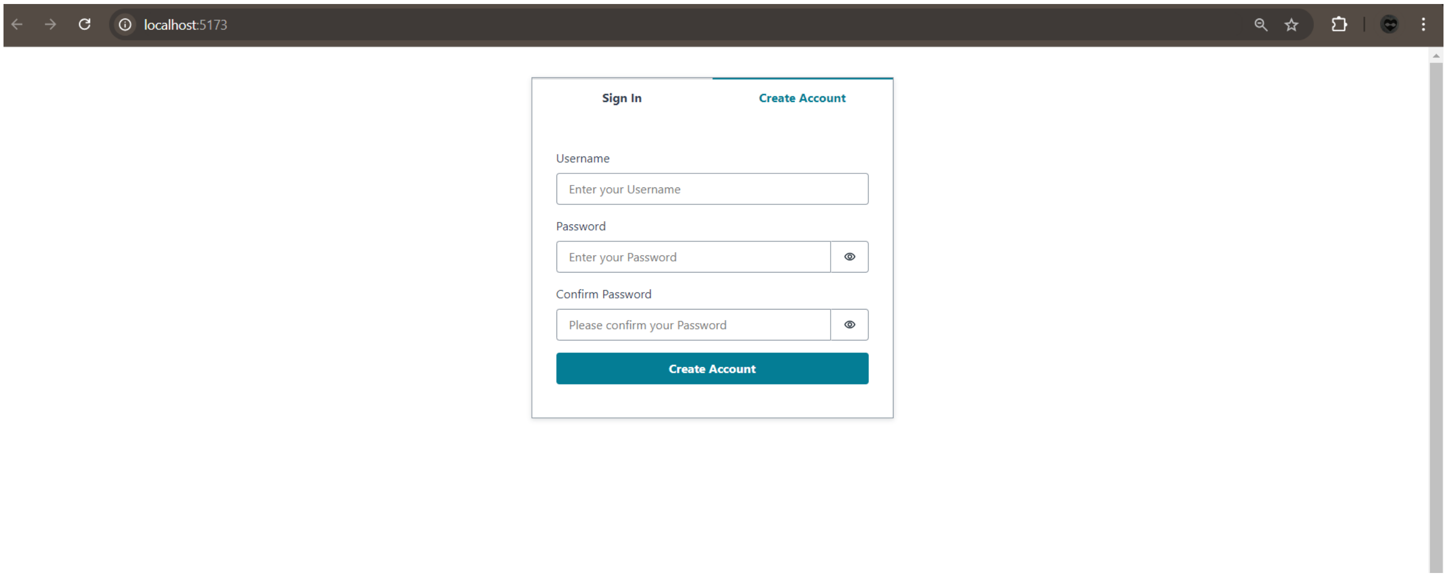
```
        alignItems="center"
        gap="2rem"
        border="1px solid #ccc"
        padding="2rem"
        borderRadius="5%"
        className="box"
       >
        <View>
         <Heading level="3">{item.title}</Heading>
        </View>
        <Text fontStyle="italic">{item.description}</Text>
        {item.image && (
         <Image
          src={item.image}
          alt={`Visual for ${item.title}`}
          style={{ width: 400 }}
         />
        )}
        <Button
         variation="destructive"
         onClick={() => deleteItem(item)}
        >
         Delete Item
        </Button>
       </Flex>
      ))}
     </Grid>
     <Button onClick={signOut}>Sign Out</Button>
    </Flex>
   )}
  </Authenticator>
 );
}
```

Launch the App Locally

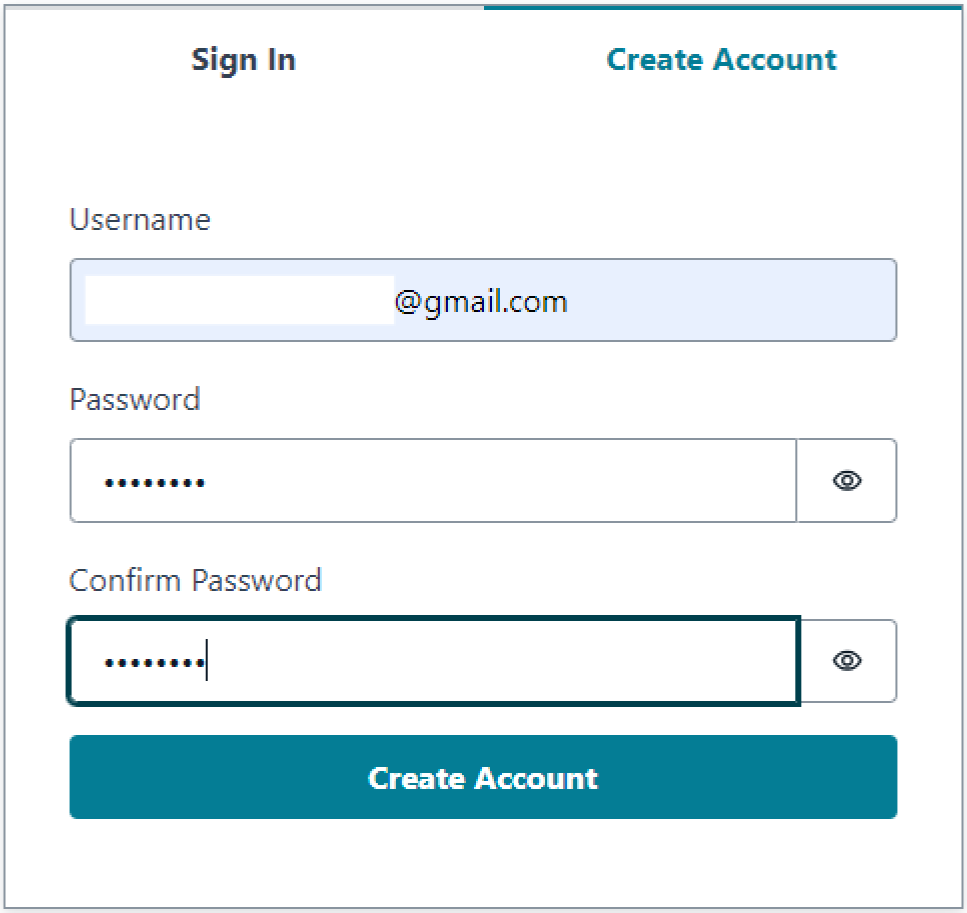
1. Open a terminal window, navigate to the root folder (bucketlistapp), and run the following command to launch the app:

npm run dev

2. Open the local host link that appears in the terminal to view the application.



3. Choose the **Create Account** tab and use the authentication flow to sign up by entering your email and password. Then, create your account.



Username

[redacted]@gmail.com

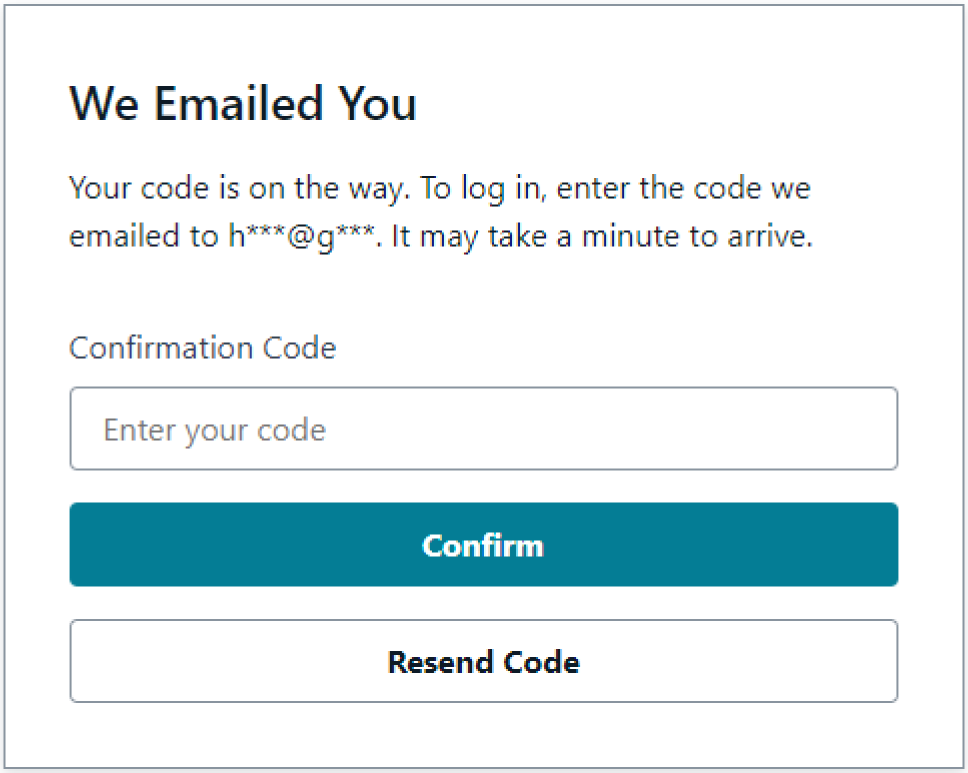
Password

••••••••

Confirm Password

••••••••

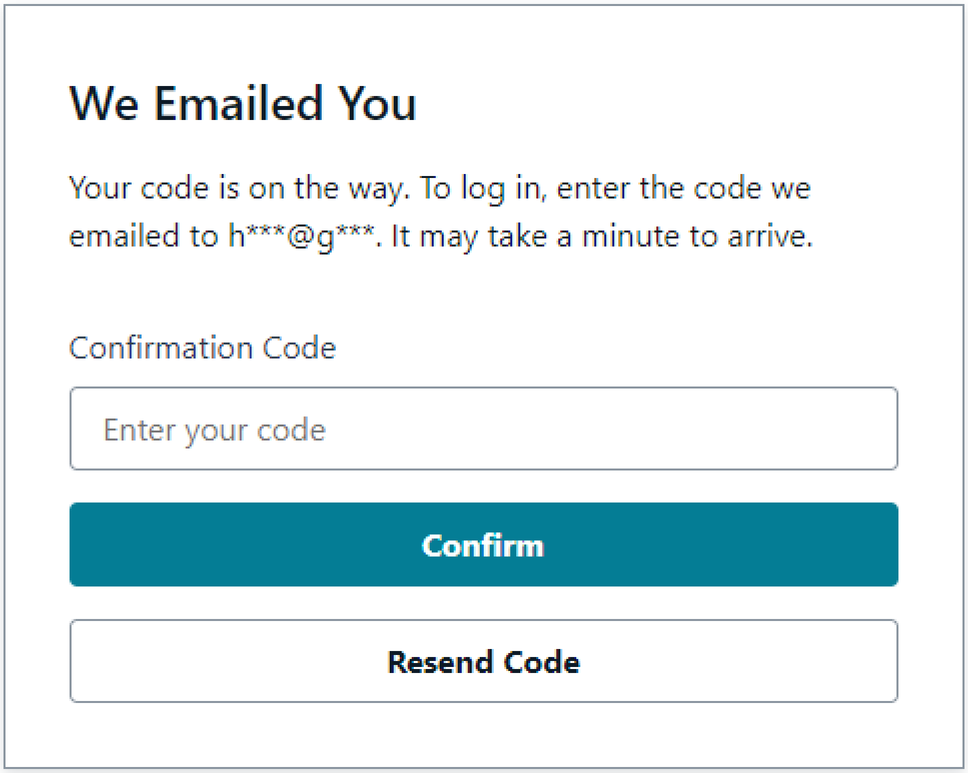
4. You will receive a verification code via email. Enter the verification code to log in.



We Emailed You

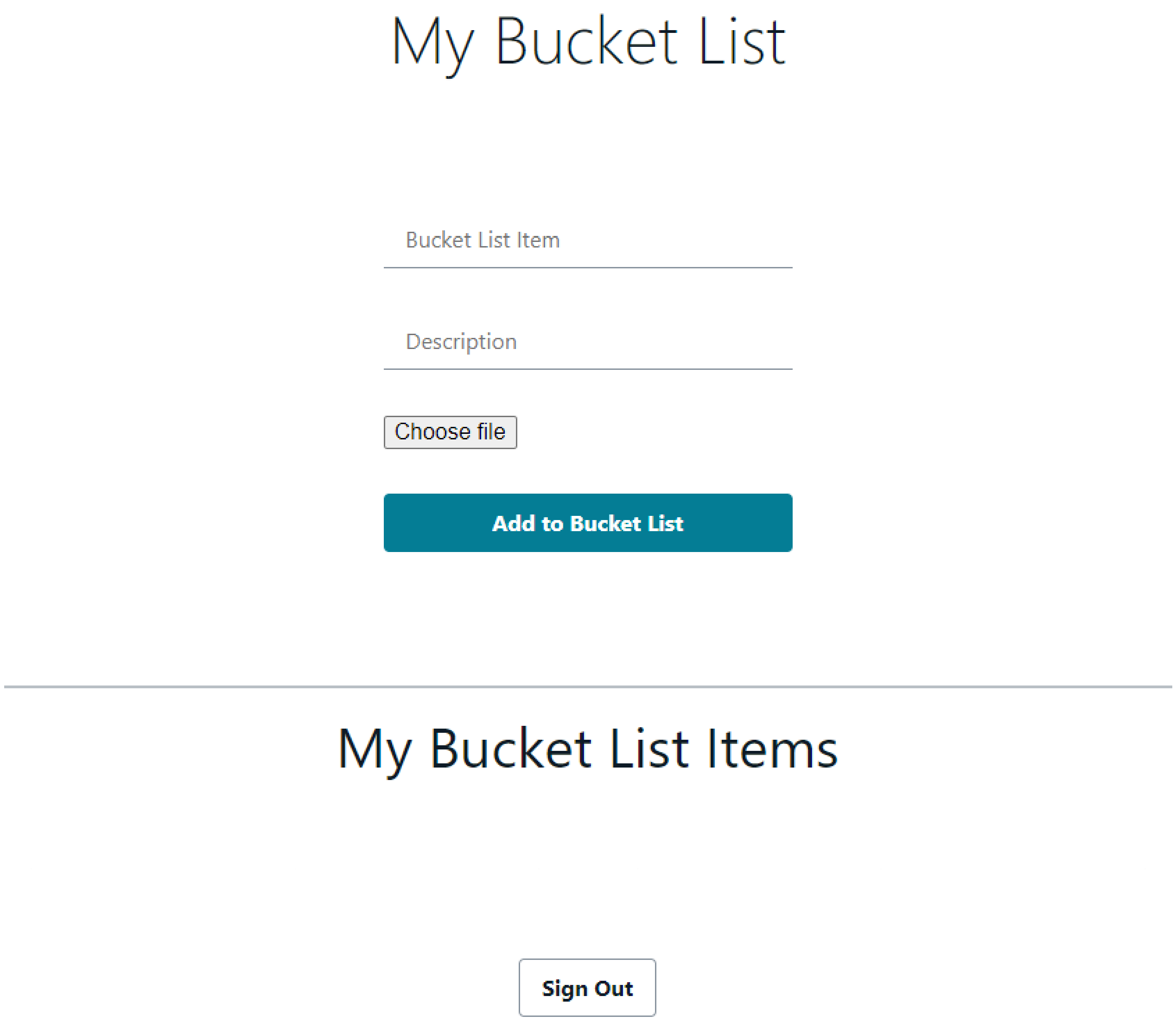
Your code is on the way. To log in, enter the code we emailed to h***@g***. It may take a minute to arrive.

5. Once signed in, you can start adding items to your bucket list and managing them.



We Emailed You

Your code is on the way. To log in, enter the code we emailed to h***@g***. It may take a minute to arrive.



Push the Changes on GitHub

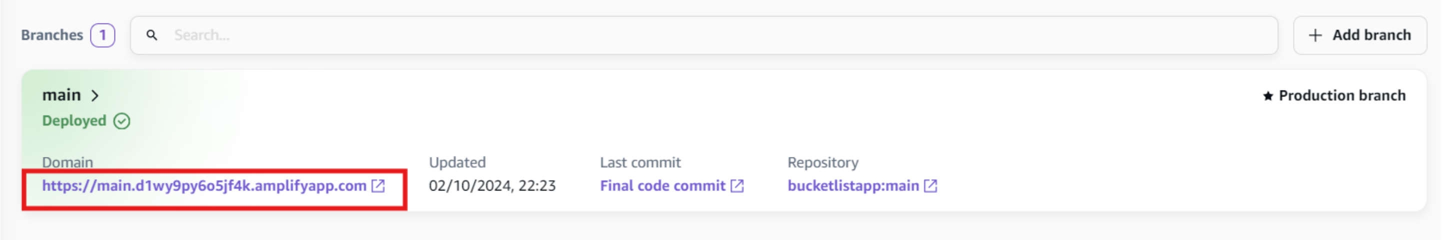
1. After making changes to the app, pushing them to GitHub with the following commands:

```
git add .

git commit -m 'bucket list tracker app'

git push origin main
```

2. Sign in to the AWS Management Console and open the [AWS Amplify console](https://console.aws.amazon.com/amplify/).

3. AWS Amplify will automatically build source code and deploying app at a URL like https://...amplifyapp.com. On every git push, Amplify will update the deployment.

4. Select the **Visit deployed URL** button to see bucket list tracker live.



main >

Deployed ✓

★ Production branch

Domain
https://main.d1wy9py6o5jf4k.amplifyapp.com

Updated
02/10/2024, 22:23

Last commit
Final code commit

Repository
bucketlistapp:main