2100031692

LADE ROHIT DATTA

URL Shortern

A step-by-step guide,

1.Sign Up for AWS:

- Go to the AWS website (https://aws.amazon.com/) and sign up for an AWS account.
- Follow the instructions to create your account. You'll need to provide payment information, but many AWS services offer a free tier for new users, which should be sufficient for this project.

2.Access AWS Management Console:

- Once your account is set up, log in to the AWS Management Console using your newly created credentials.

3.Navigate to DynamoDB:

- In the AWS Management Console, find the "Services" dropdown menu at the top left.
- Under the "Databases" section, select "DynamoDB."

4.Create a DynamoDB Table:

- In the DynamoDB console, click on the "Create table" button.
- Choose a name for your table (e.g., "URLShortener").
- Define the primary key for your table. For simplicity, you can use a single attribute called "ShortURL" or "Key" with a data type of String.
- Click "Create" to create the table.

5.Navigate to Lambda:

- From the AWS Management Console, go to the "Services" dropdown menu.
- Under the "Compute" section, select "Lambda."

6.Create a Lambda Function:

- Click on the "Create function" button.

- Choose the option to author from scratch.

- Give your function a name and select a runtime (e.g., Node.js, Python).

- For this project, you can start with a simple Node.js or Python function.

- Write the code for your Lambda function to handle URL shortening and redirection logic. You'll need to interact with the DynamoDB table you created earlier.

7.Navigate to API Gateway:

- From the AWS Management Console, go to the "Services" dropdown menu.
- Under the "Networking & Content Delivery" section, select "API Gateway."

8.Create an API:

- Click on the "Create API" button.
- Choose the option to build a REST API.
- Create a new resource (e.g., "/shorten") and a corresponding POST method. This will be used for shortening URLs.
- Create another resource (e.g., "/{shortURL}") with a GET method. This will be used for redirecting shortened URLs.

9.Integrate API with Lambda:

- For both the POST and GET methods, integrate them with the Lambda function you created earlier.
- Set up the appropriate mapping templates for request and response payloads.

10.Deploy the API:

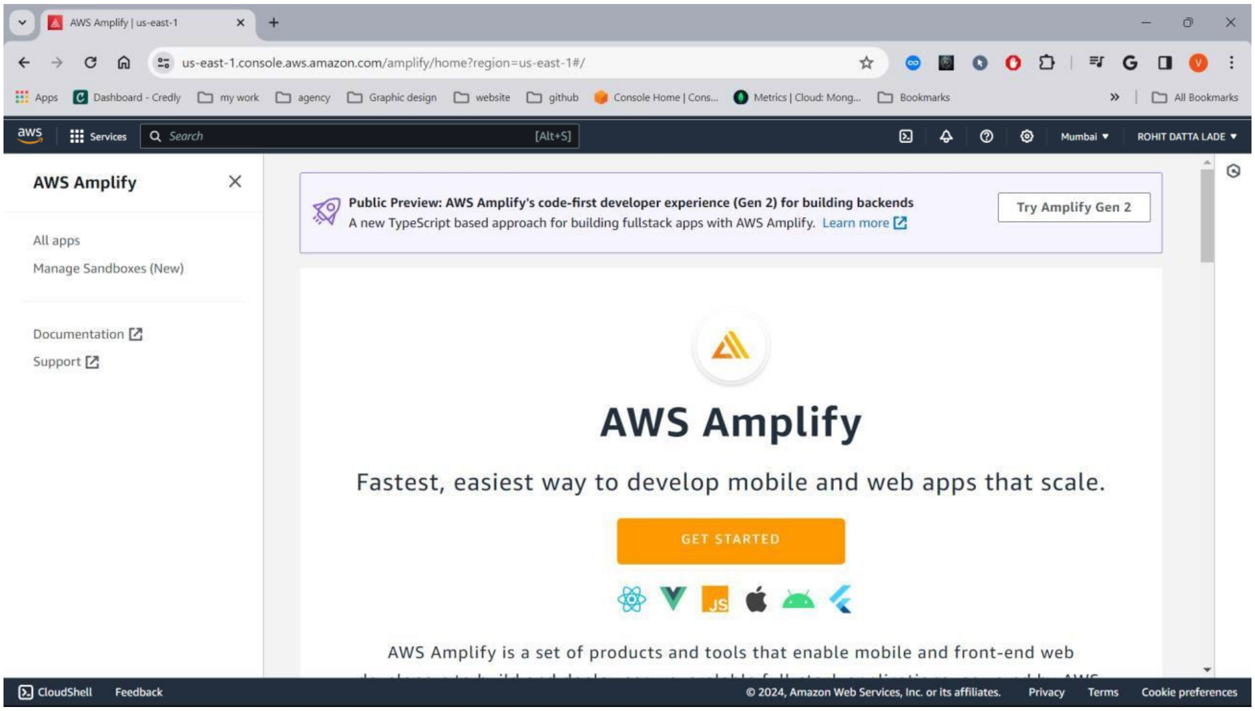
- Once your API is configured, deploy it to make it accessible over the internet.

- You'll get a unique endpoint URL that you can use to interact with your URL shortener.

11. Testing:

- Test your API using tools like Postman or cURL. Send POST requests to shorten URLs and GET requests to access shortened URLs and verify redirection.

Screen shots:



AWS Amplify

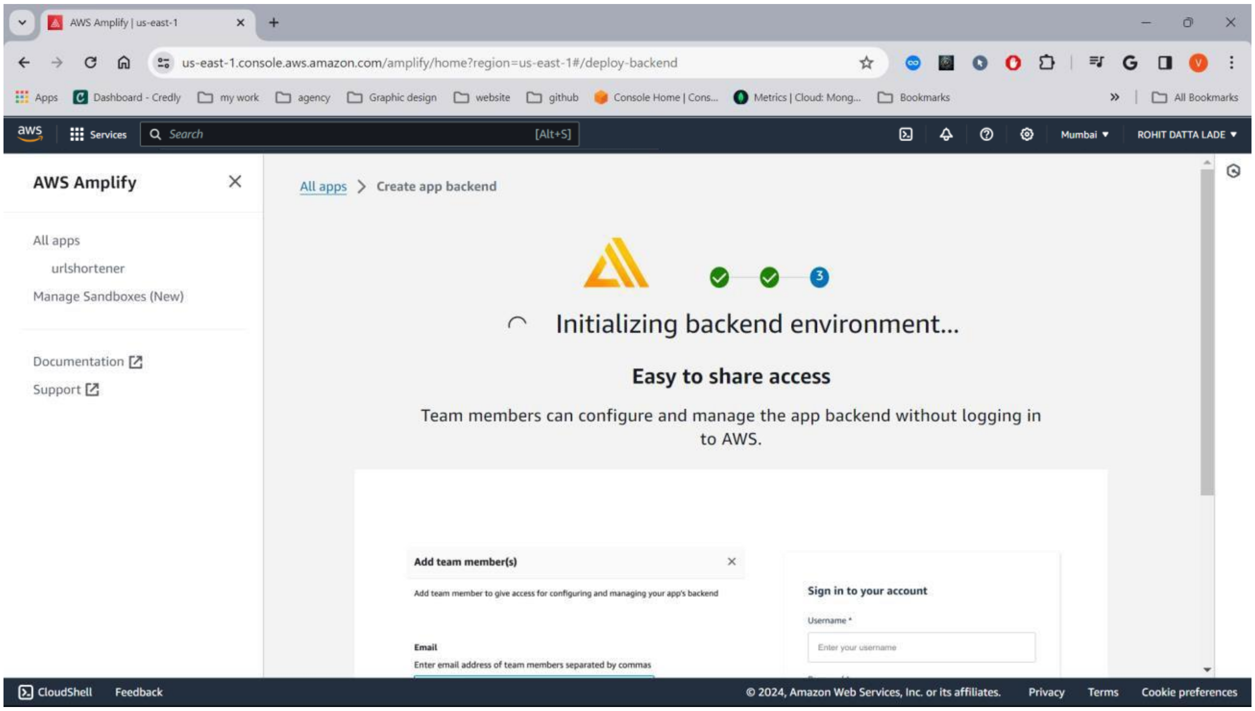
Public Preview: AWS Amplify's code-first developer experience (Gen 2) for building backends
A new TypeScript based approach for building fullstack apps with AWS Amplify. Learn more

AWS Amplify

Fastest, easiest way to develop mobile and web apps that scale.

AWS Amplify is a set of products and tools that enable mobile and front-end web

© 2024, Amazon Web Services, Inc. or its affiliates.



AWS Amplify

All apps > Create app backend

Initializing backend environment...

Easy to share access

Team members can configure and manage the app backend without logging in to AWS.

Add team member(s)

Add team member to give access for configuring and managing your app's backend

Email

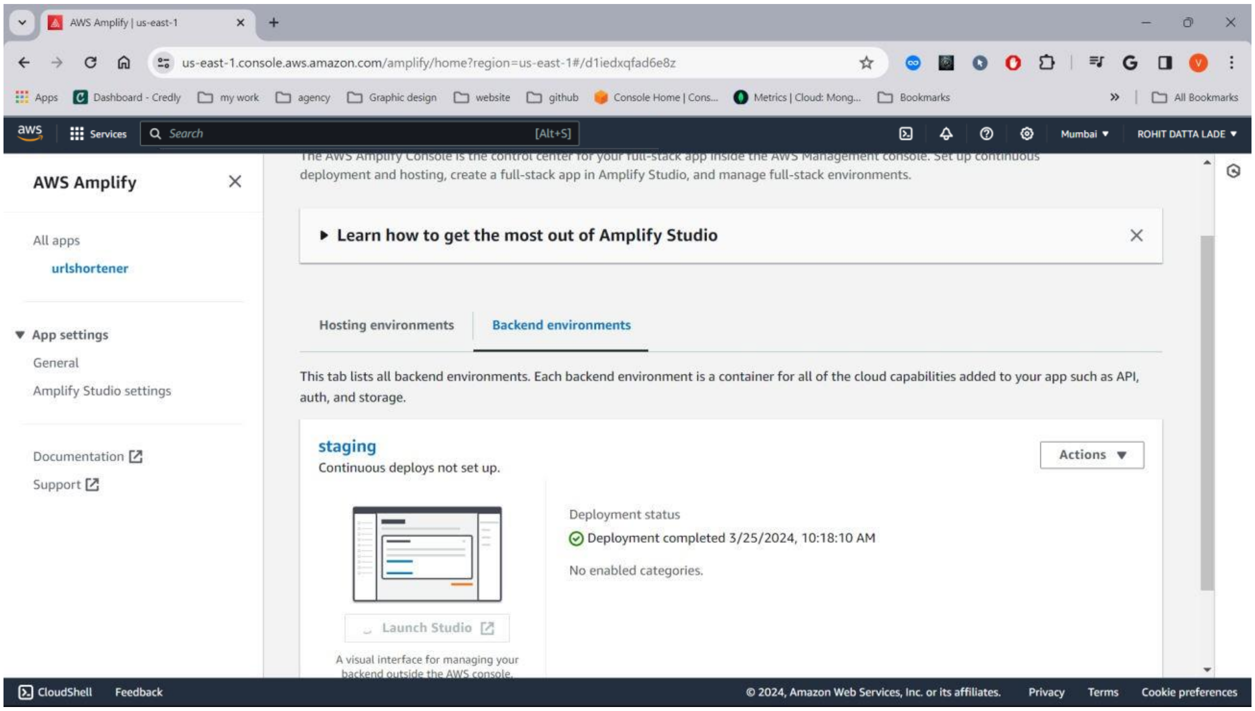
Enter email address of team members separated by commas

Sign in to your account

Username *

Enter your username

© 2024, Amazon Web Services, Inc. or its affiliates.



The AWS Amplify Console is the control center for your full-stack app inside the AWS Management console. Set up continuous deployment and hosting, create a full-stack app in Amplify Studio, and manage full-stack environments.

▸ **Learn how to get the most out of Amplify Studio**

This tab lists all backend environments. Each backend environment is a container for all of the cloud capabilities added to your app such as API, auth, and storage.

staging

Continuous deploys not set up.

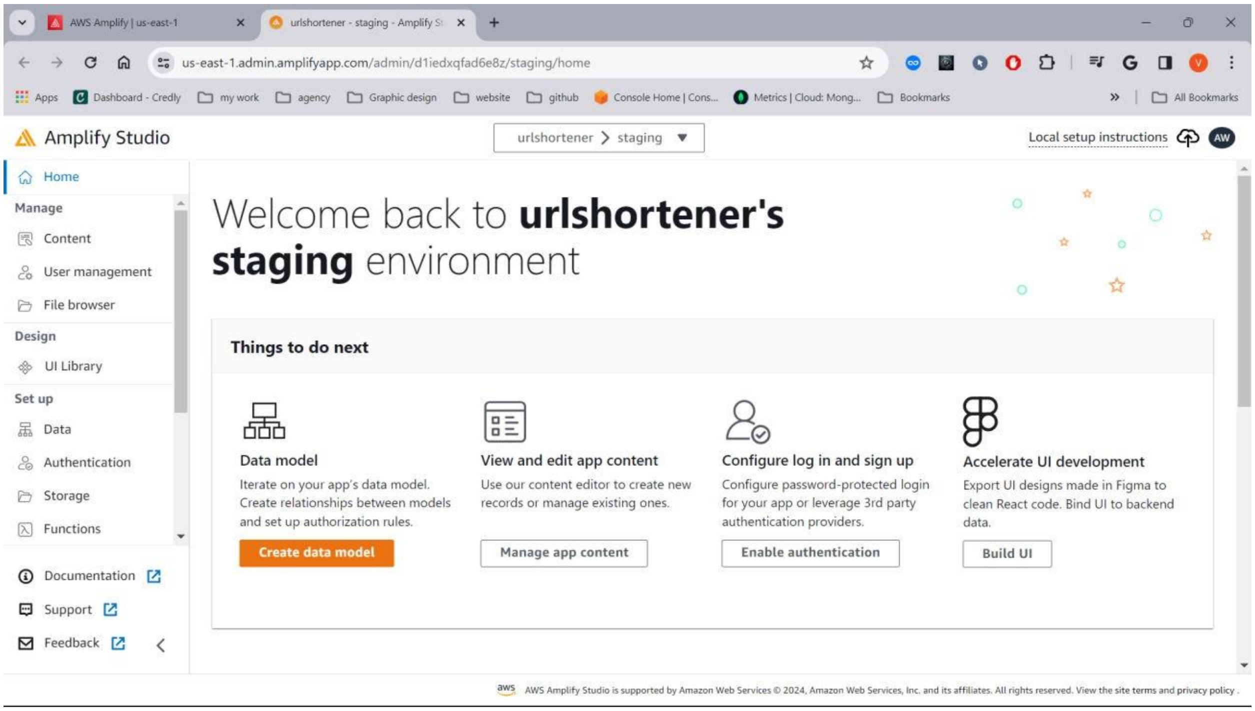
Deployment status

✓ Deployment completed 3/25/2024, 10:18:10 AM

No enabled categories.

A visual interface for managing your backend outside the AWS console.

© 2024, Amazon Web Services, Inc. or its affiliates.



Welcome back to **urlshortener's staging** environment

Things to do next

Data model

Iterate on your app's data model. Create relationships between models and set up authorization rules.

View and edit app content

Use our content editor to create new records or manage existing ones.

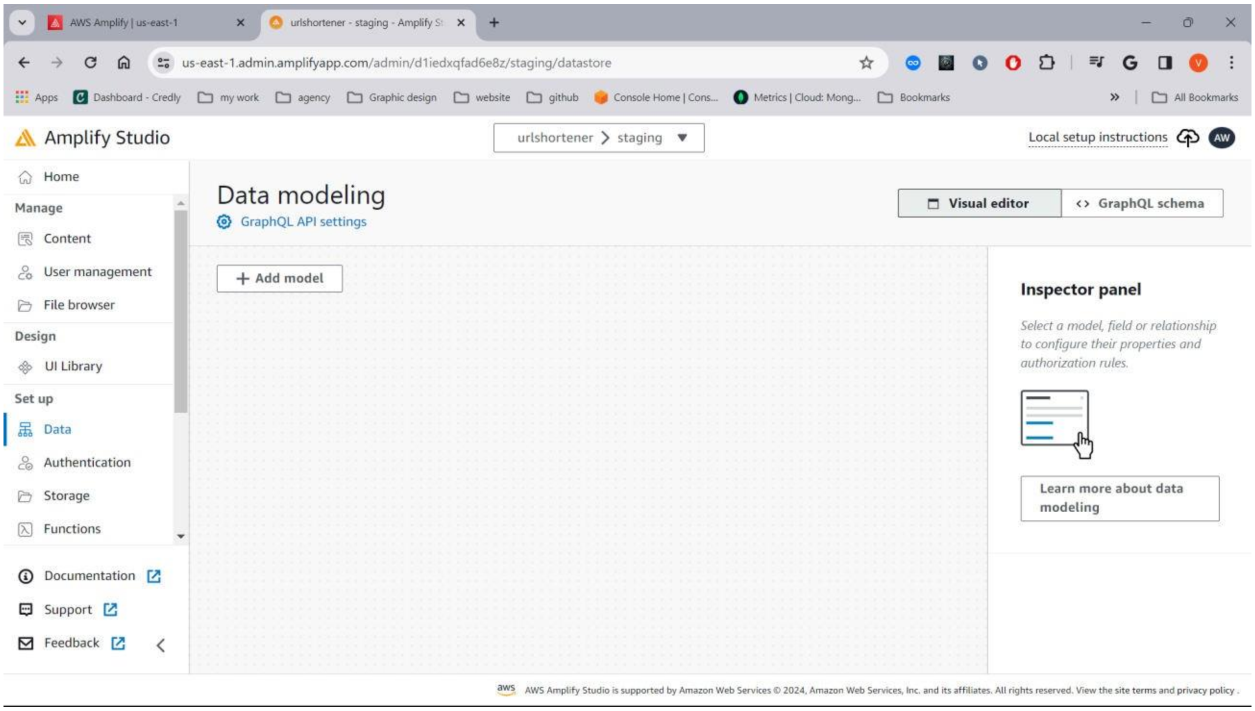
Configure log in and sign up

Configure password-protected login for your app or leverage 3rd party authentication providers.

Accelerate UI development

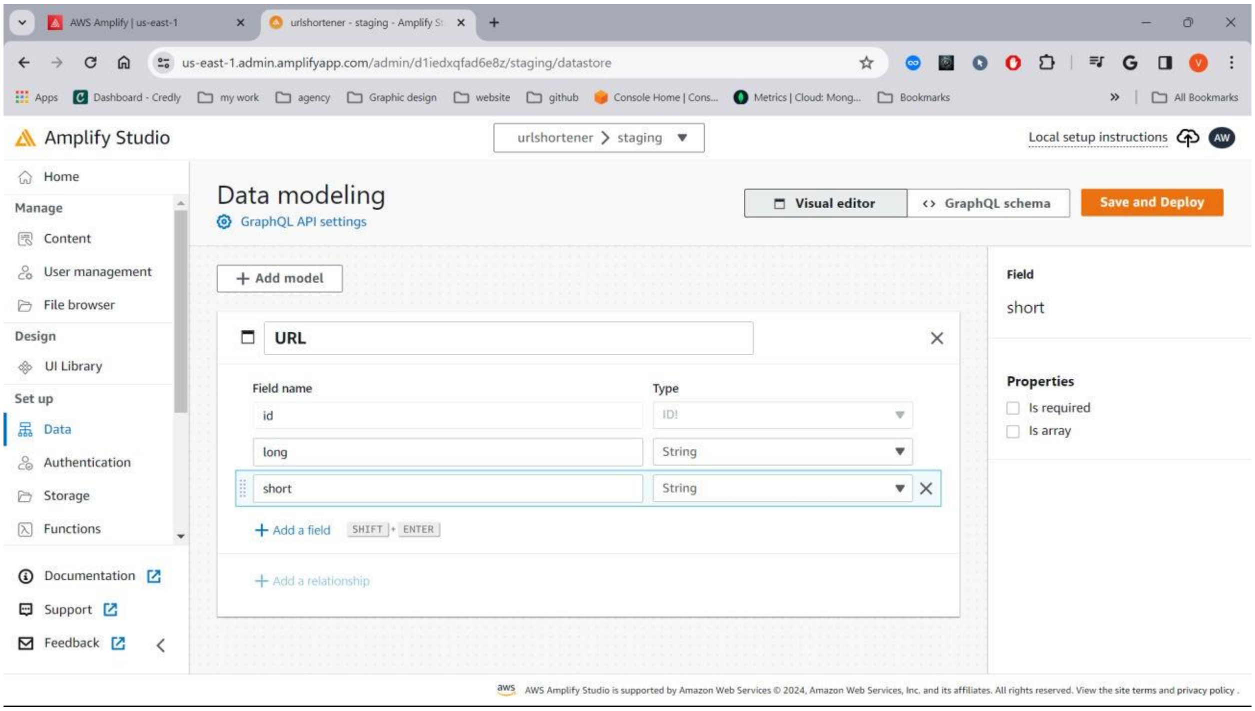
Export UI designs made in Figma to clean React code. Bind UI to backend data.

AWS Amplify Studio is supported by Amazon Web Services © 2024, Amazon Web Services, Inc. and its affiliates. All rights reserved. View the site terms and privacy policy .



Select a model, field or relationship to configure their properties and authorization rules.

AWS Amplify Studio is supported by Amazon Web Services © 2024, Amazon Web Services, Inc. and its affiliates. All rights reserved. View the site terms and privacy policy .



URL

Field name	Type
id	ID!
long	String
short	String

Field

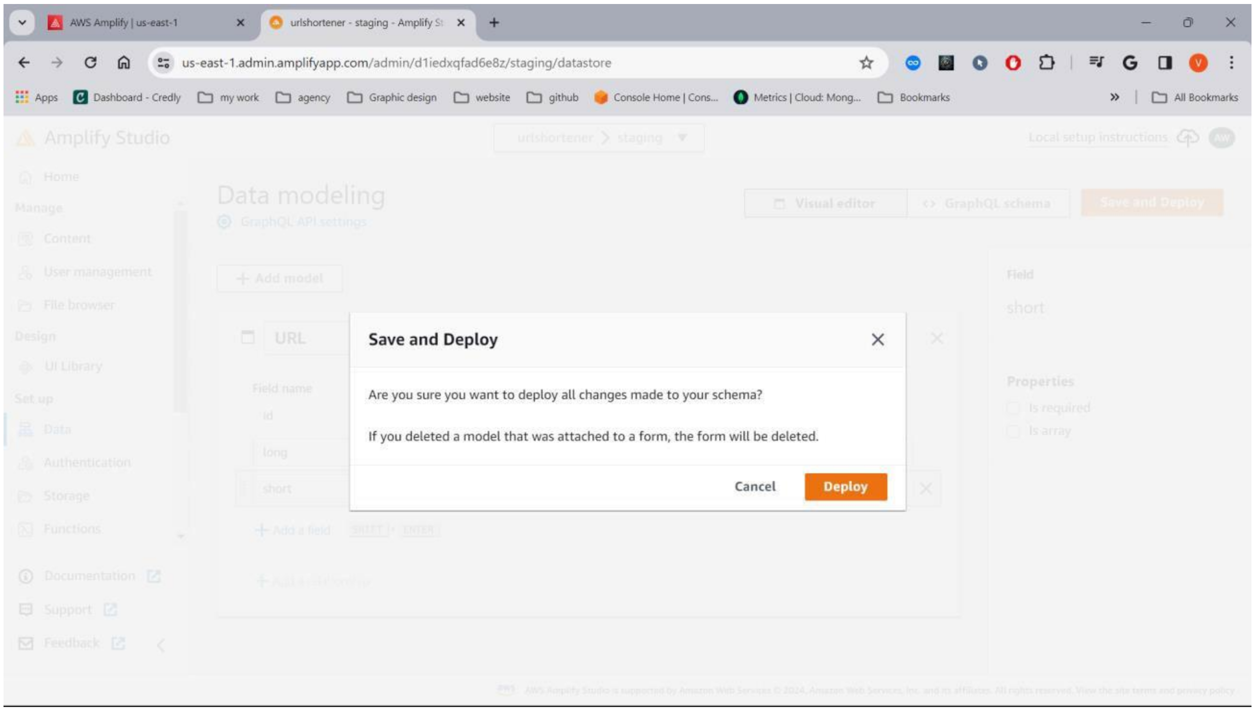
short

Properties

□ Is required

□ Is array

AWS Amplify Studio is supported by Amazon Web Services © 2024, Amazon Web Services, Inc. and its affiliates. All rights reserved. View the site terms and privacy policy .

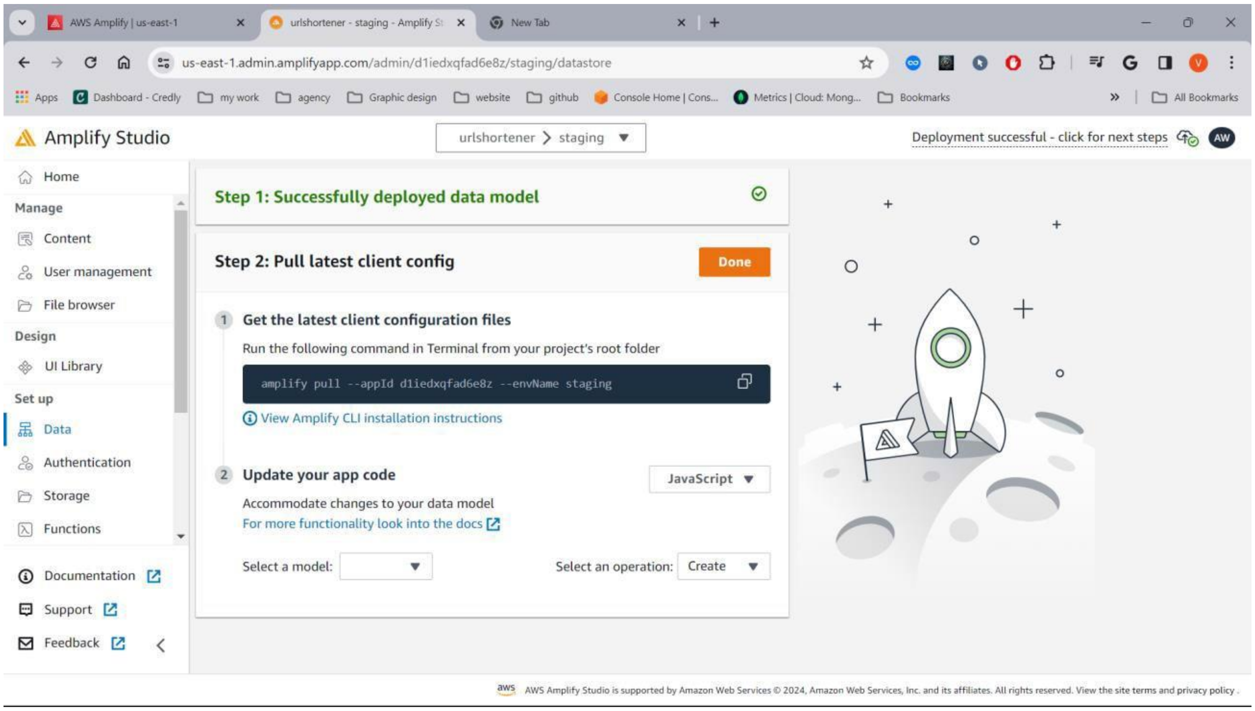


Save and Deploy

Are you sure you want to deploy all changes made to your schema?

If you deleted a model that was attached to a form, the form will be deleted.

Cancel Deploy



Deployment successful - click for next steps

Step 1: Successfully deployed data model

Step 2: Pull latest client config Done

1 **Get the latest client configuration files**

Run the following command in Terminal from your project's root folder

```
amplify pull --appId d1iedxqfad6e8z --envName staging
```

View Amplify CLI installation instructions

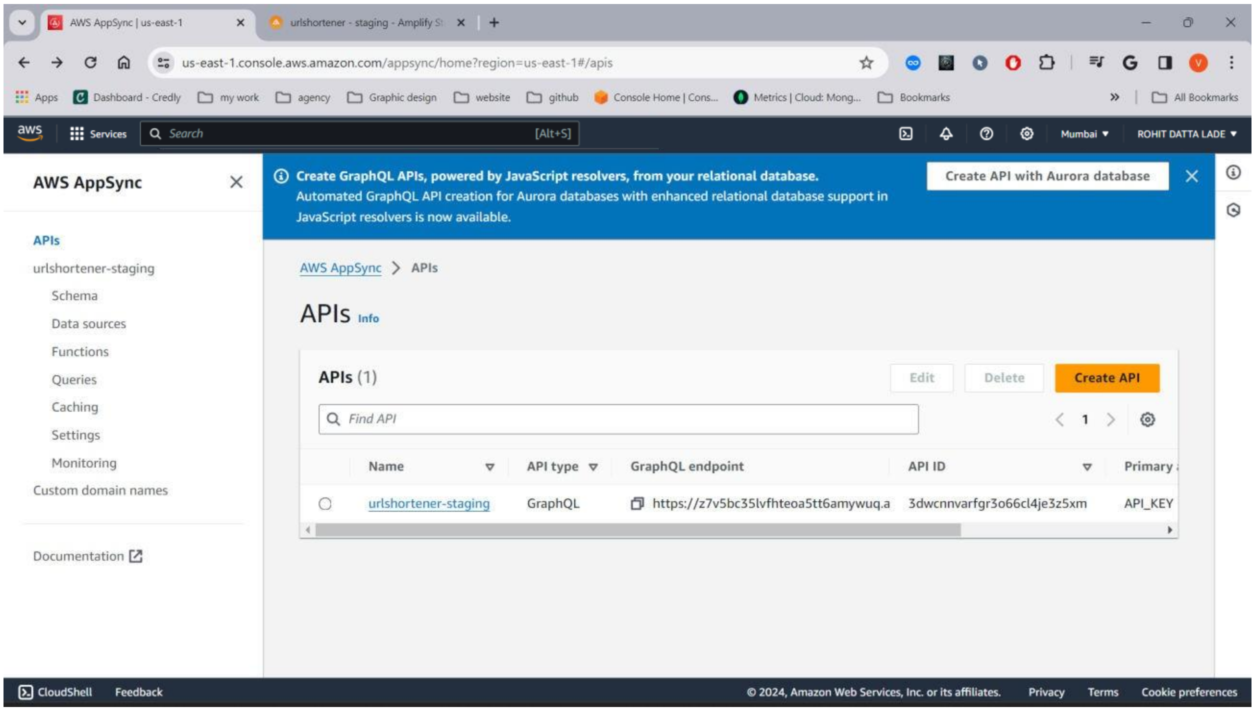
2 **Update your app code** JavaScript

Accommodate changes to your data model

For more functionality look into the docs

Select a model: Select an operation: Create

AWS Amplify Studio is supported by Amazon Web Services © 2024, Amazon Web Services, Inc. and its affiliates. All rights reserved. View the site terms and privacy policy .

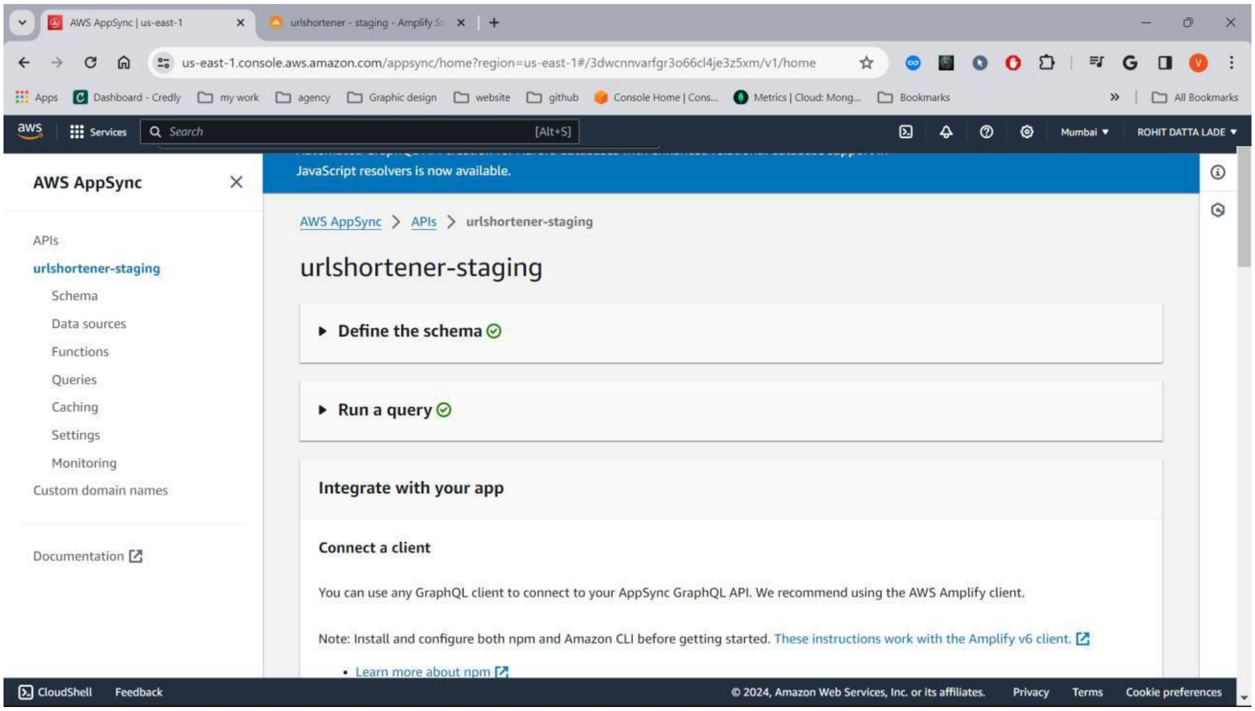


Create GraphQL APIs, powered by JavaScript resolvers, from your relational database.
Automated GraphQL API creation for Aurora databases with enhanced relational database support in JavaScript resolvers is now available.

APIs

APIs (1)

	Name	API type	GraphQL endpoint	API ID	Primary a
○	urlshortener-staging	GraphQL	https://z7v5bc35lvfhteoa5tt6amywuq.a	3dwcnnvarfgr3o66cl4je3z5xm	API_KEY



JavaScript resolvers is now available.

urlshortener-staging

▸ Define the schema ✓

▸ Run a query ✓

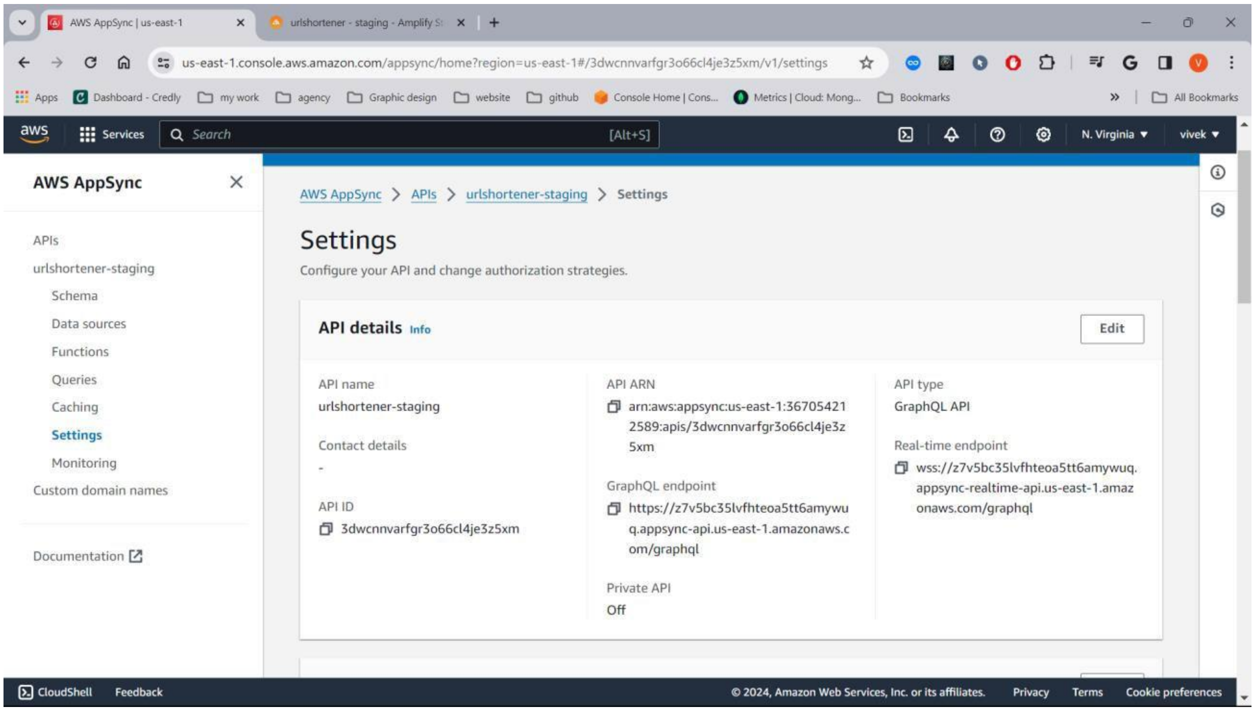
Integrate with your app

Connect a client

You can use any GraphQL client to connect to your AppSync GraphQL API. We recommend using the AWS Amplify client.

Note: Install and configure both npm and Amazon CLI before getting started. These instructions work with the Amplify v6 client.

- Learn more about npm



Settings

Configure your API and change authorization strategies.

API details Info

API name
urlshortener-staging

Contact details
-

API ID
3dwcnnvarfgr3o66cl4je3z5xm

API ARN
arn:aws:appsync:us-east-1:367054212589:apis/3dwcnnvarfgr3o66cl4je3z5xm

GraphQL endpoint
https://z7v5bc35lvfhteoa5tt6amywuq.appsync-api.us-east-1.amazonaws.com/graphql

Private API
Off

API type
GraphQL API

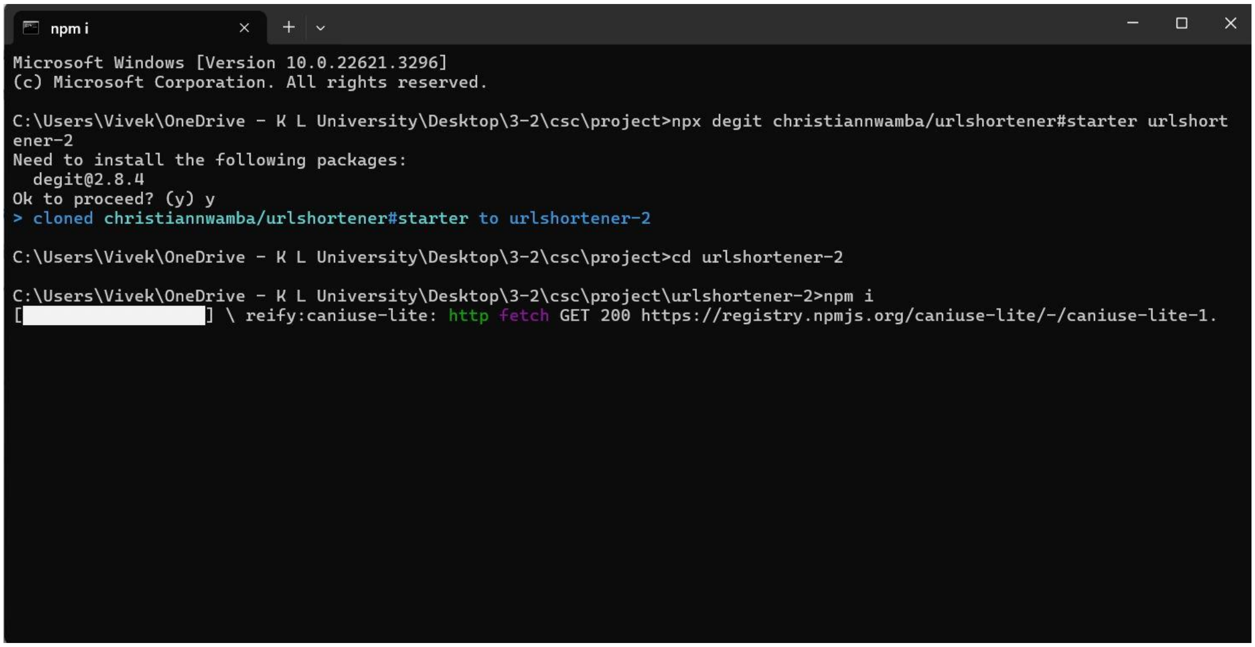
Real-time endpoint
wss://z7v5bc35lvfhteoa5tt6amywuq.appsync-realtime-api.us-east-1.amazonaws.com/graphql

endpoint

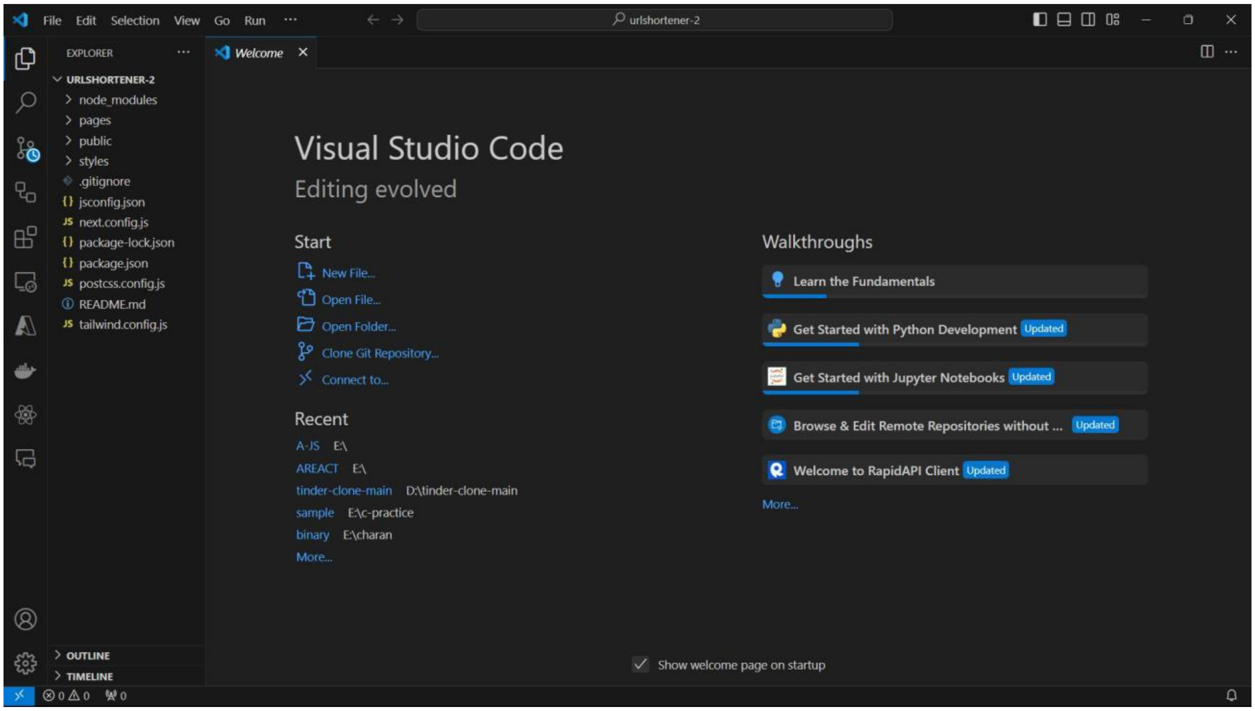
https://z7v5bc35lvfhteoa5tt6amywuq.appsync-api.us-east-1.amazonaws.com/graphql

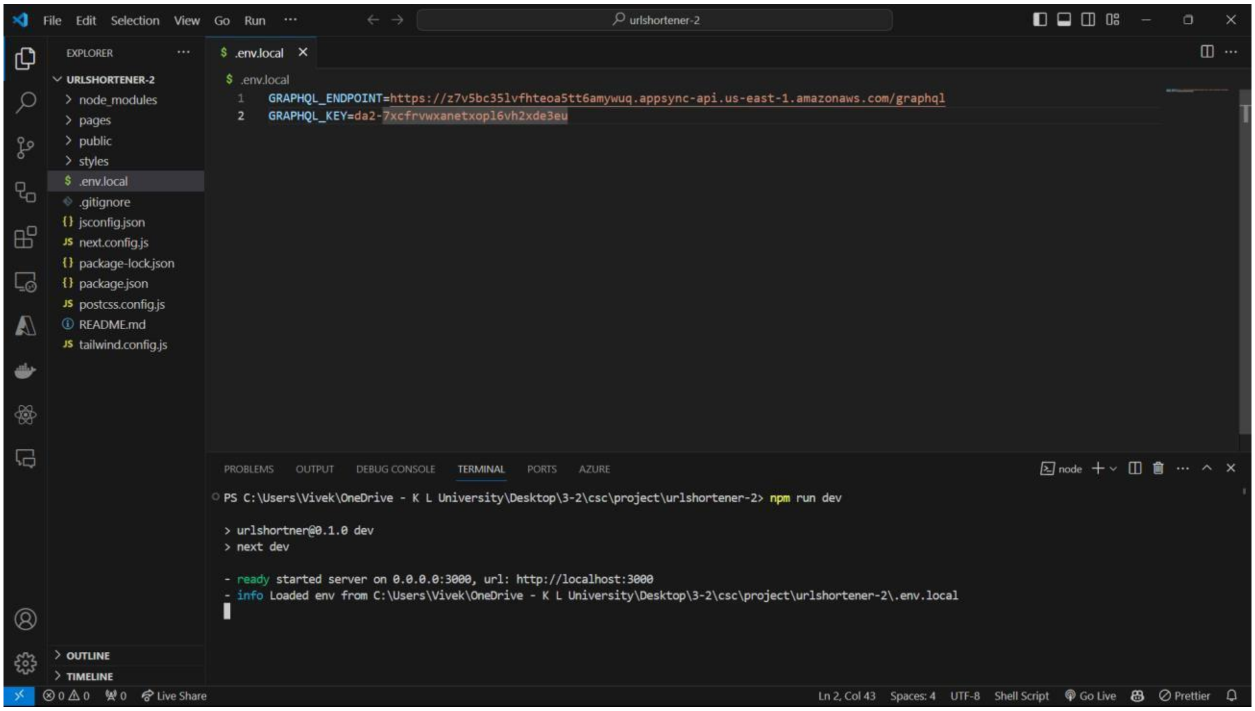
api key

da2-7xcfrvwxanetxopl6vh2xde3eu



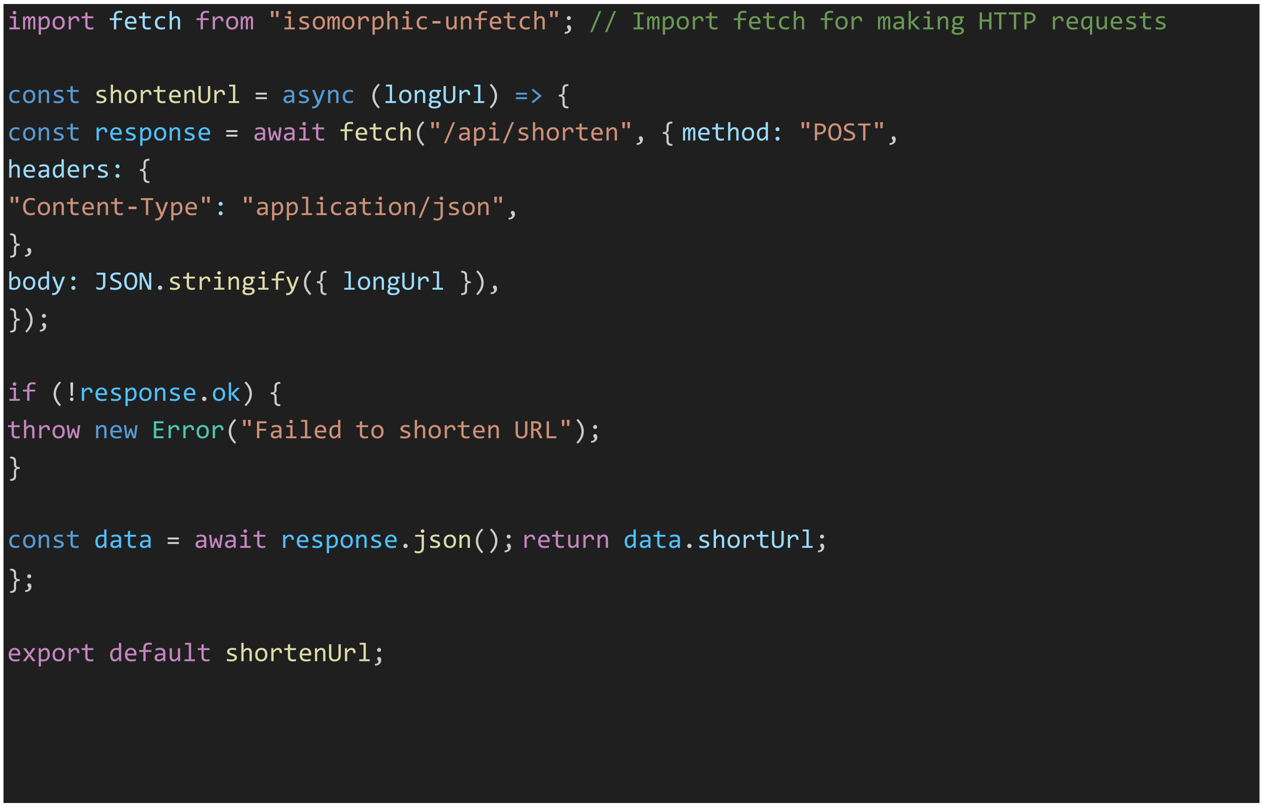
```
Microsoft Windows [Version 10.0.22621.3296]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Vivek\OneDrive - K L University\Desktop\3-2\csc\project>npx degit christiannwamba/urlshortener#starter urlshort
ener-2
Need to install the following packages:
  degit@2.8.4
Ok to proceed? (y) y
> cloned christiannwamba/urlshortener#starter to urlshortener-2

C:\Users\Vivek\OneDrive - K L University\Desktop\3-2\csc\project>cd urlshortener-2

C:\Users\Vivek\OneDrive - K L University\Desktop\3-2\csc\project\urlshortener-2>npm i
[██████████████████] \ reify:caniuse-lite: http fetch GET 200 https://registry.npmjs.org/caniuse-lite/-/caniuse-lite-1.
```



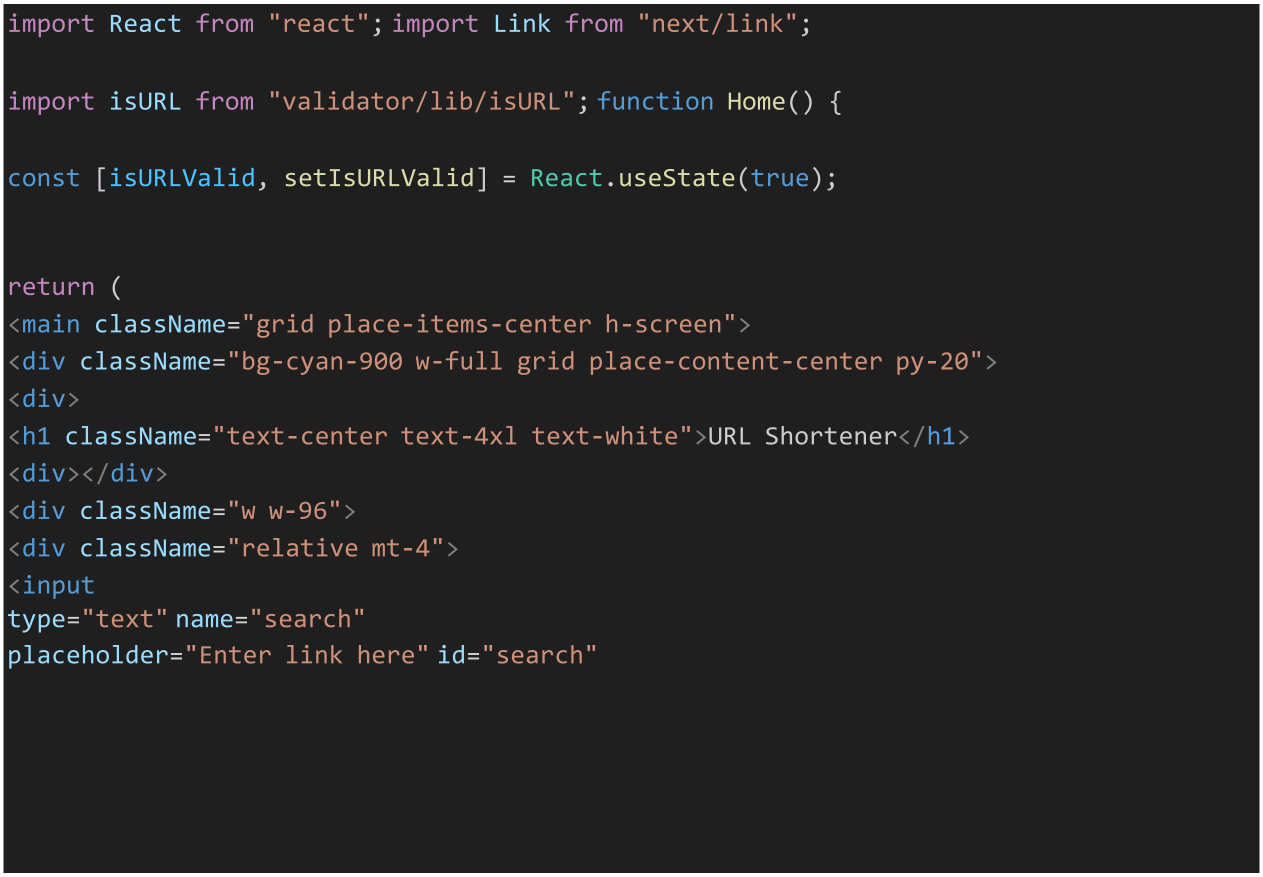


```
GRAPHQL_ENDPOINT=https://z7v5bc35lvfhteoa5tt6amywuq.appsync-api.us-east-1.amazonaws.com/graphql
GRAPHQL_KEY=da2-7xcfrvwxanetxopl6vh2xde3eu
```

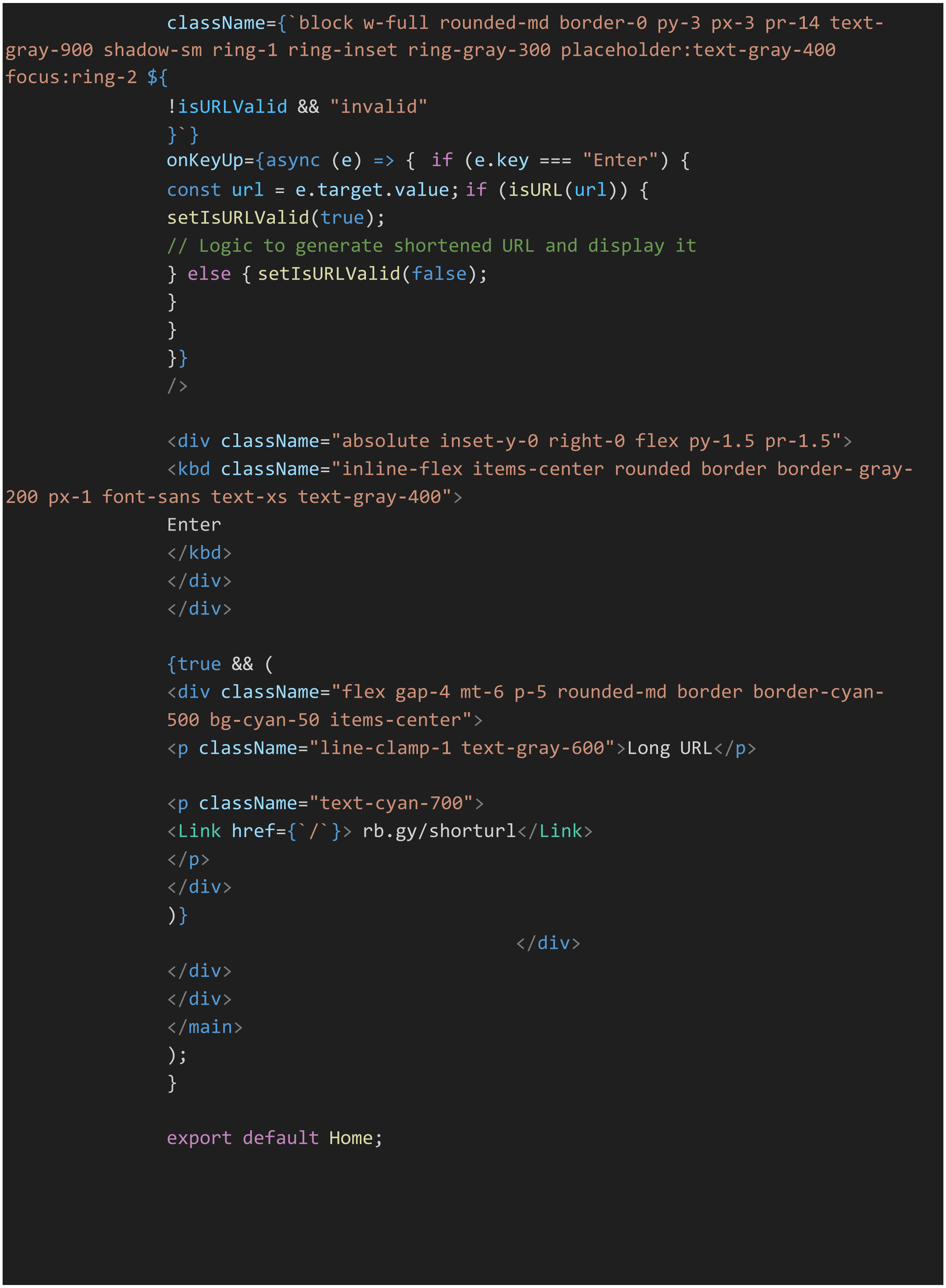
```
PS C:\Users\Vivek\OneDrive - K L University\Desktop\3-2\csc\project\urlshortener-2> npm run dev

> urlshortner@0.1.0 dev
> next dev

- ready started server on 0.0.0.0:3000, url: http://localhost:3000
- info Loaded env from C:\Users\Vivek\OneDrive - K L University\Desktop\3-2\csc\project\urlshortener-2\.env.local
```



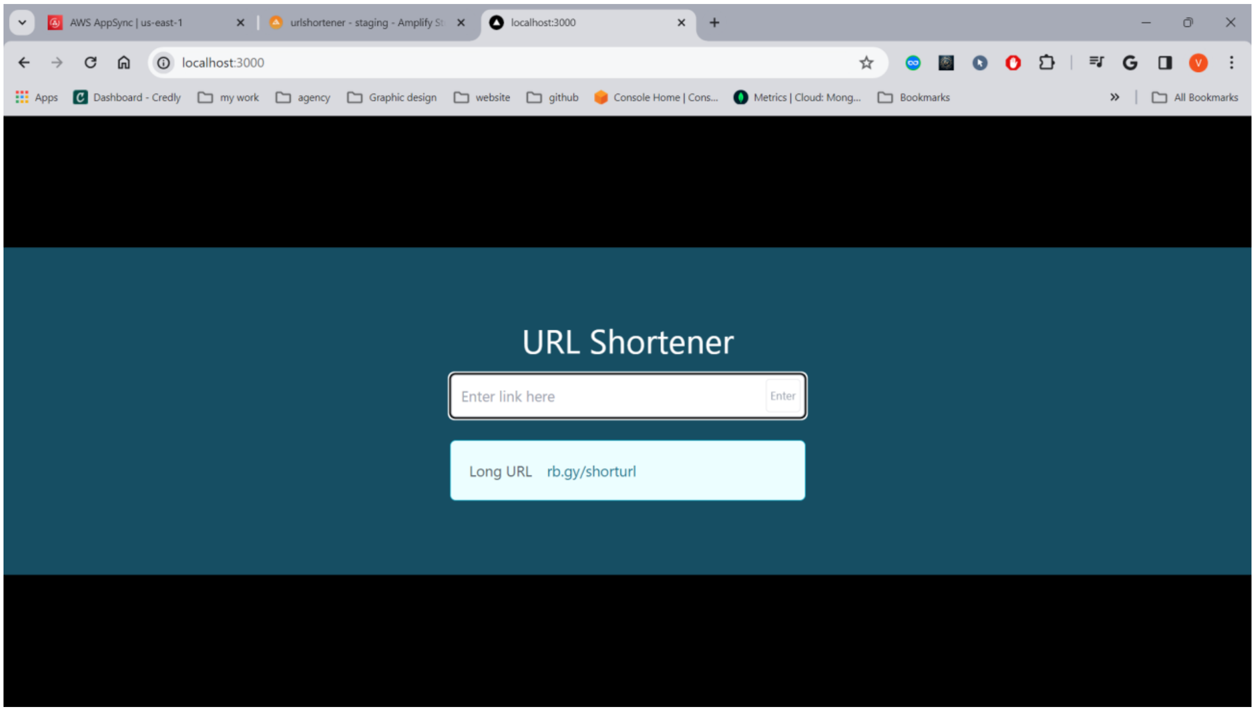
```
import fetch from "isomorphic-unfetch"; // Import fetch for making HTTP requests

const shortenUrl = async (longUrl) => {
const response = await fetch("/api/shorten", { method: "POST",
headers: {
"Content-Type": "application/json",
},
body: JSON.stringify({ longUrl }),
});

if (!response.ok) {
throw new Error("Failed to shorten URL");
}

const data = await response.json(); return data.shortUrl;
};

export default shortenUrl;
```



```
import React from "react"; import Link from "next/link";

import isURL from "validator/lib/isURL"; function Home() {

const [isURLValid, setIsURLValid] = React.useState(true);


return (
<main className="grid place-items-center h-screen">
<div className="bg-cyan-900 w-full grid place-content-center py-20">
<div>
<h1 className="text-center text-4xl text-white">URL Shortener</h1>
<div></div>
<div className="w w-96">
<div className="relative mt-4">
<input
type="text" name="search"
placeholder="Enter link here" id="search"
```

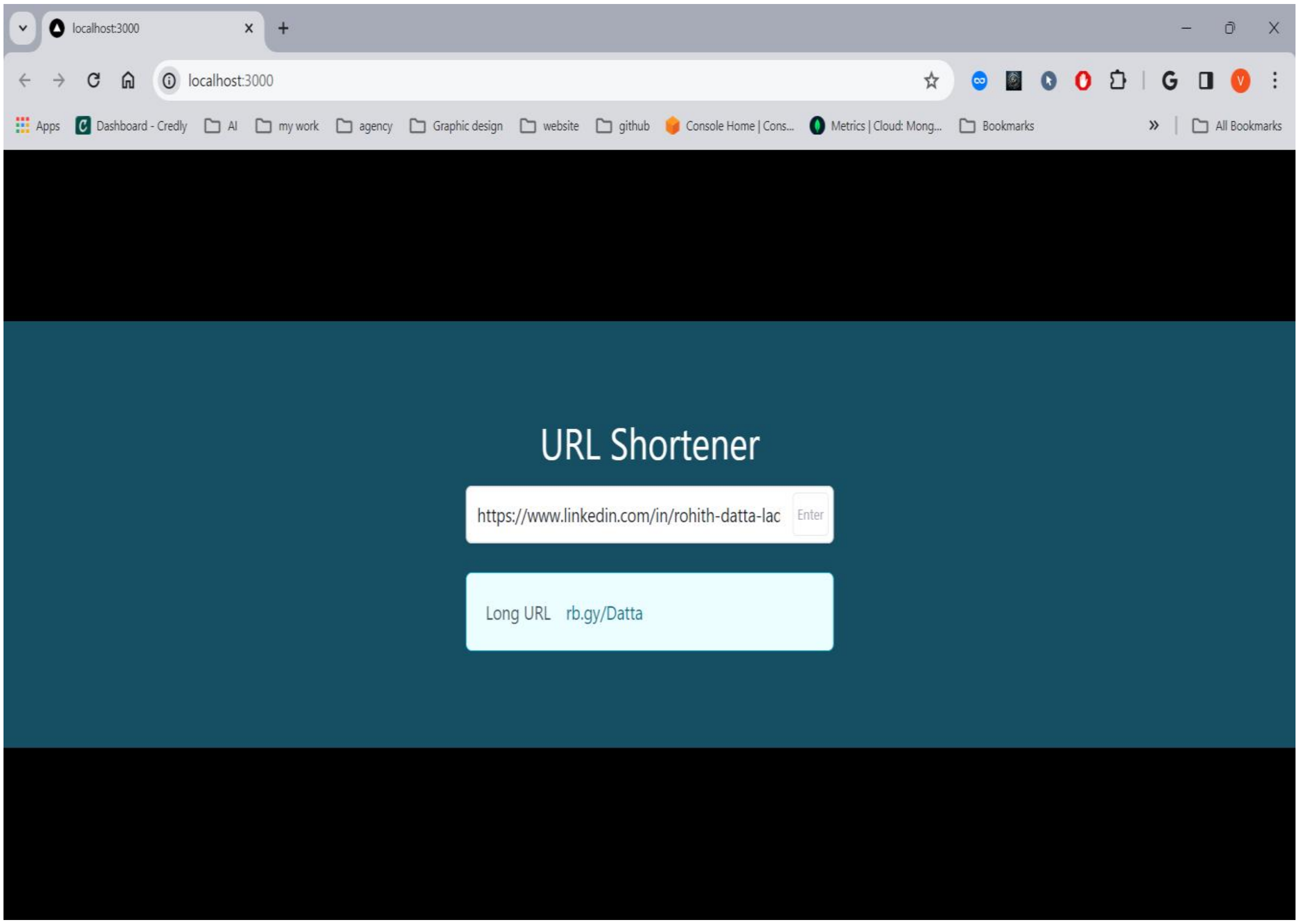


```
className={`block w-full rounded-md border-0 py-3 px-3 pr-14 text-gray-900 shadow-sm ring-1 ring-inset ring-gray-300 placeholder:text-gray-400 focus:ring-2 ${
!isURLValid && "invalid"
}`}
onKeyUp={async (e) => {  if (e.key === "Enter") {
const url = e.target.value; if (isURL(url)) {
setIsURLValid(true);
// Logic to generate shortened URL and display it
} else { setIsURLValid(false);
}
}
}}
/>

<div className="absolute inset-y-0 right-0 flex py-1.5 pr-1.5">
<kbd className="inline-flex items-center rounded border border- gray-200 px-1 font-sans text-xs text-gray-400">
Enter
</kbd>
</div>
</div>

{true && (
<div className="flex gap-4 mt-6 p-5 rounded-md border border-cyan-500 bg-cyan-50 items-center">
<p className="line-clamp-1 text-gray-600">Long URL</p>

<p className="text-cyan-700">
<Link href={`/`}> rb.gy/shorturl</Link>
</p>
</div>
)}
                                </div>
</div>
</div>
</main>
);
}

export default Home;
```



URL Shortener

Long URL rb.gy/shorturl



URL Shortener

https://www.linkedin.com/in/rohith-datta-lac

Long URL rb.gy/Datta