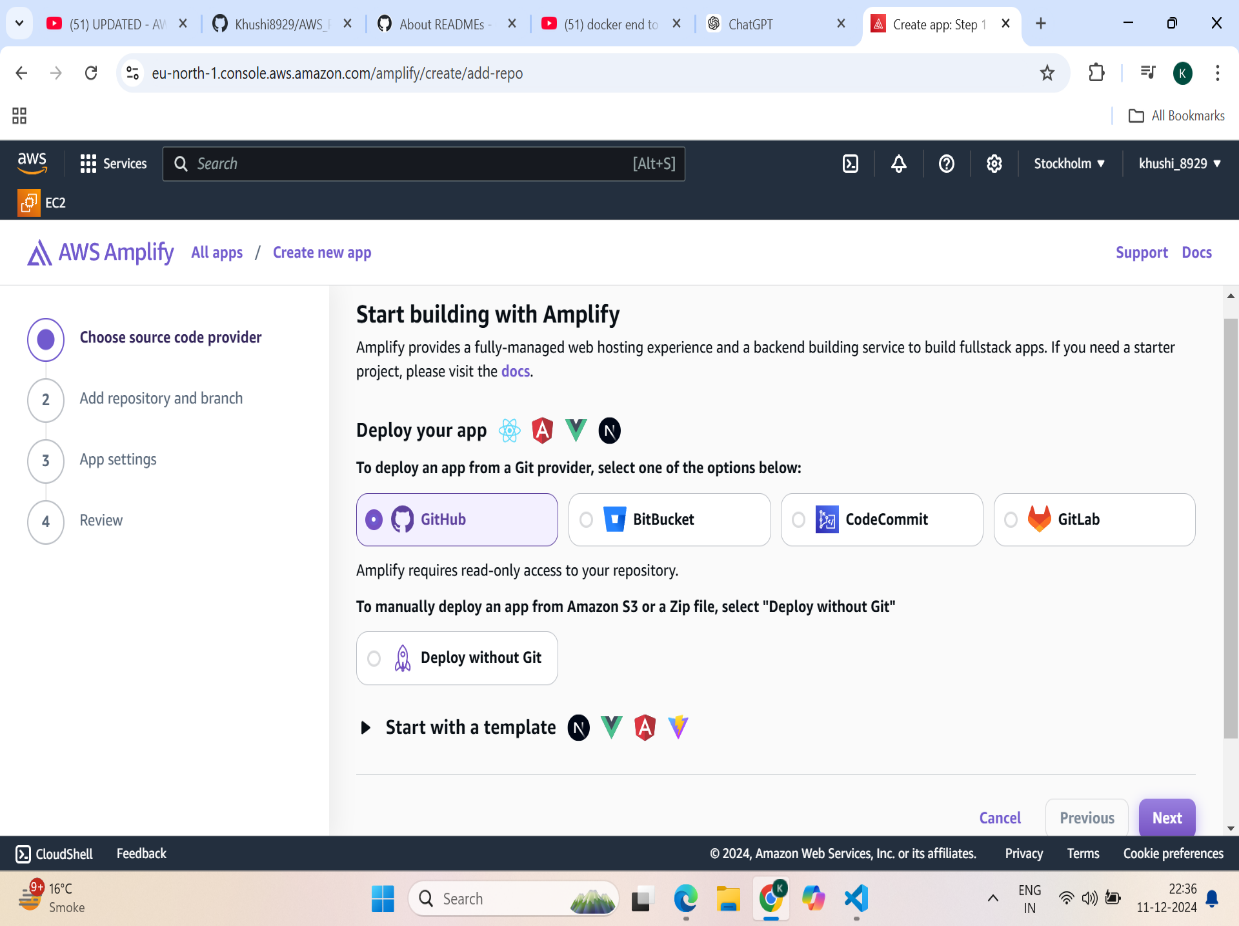
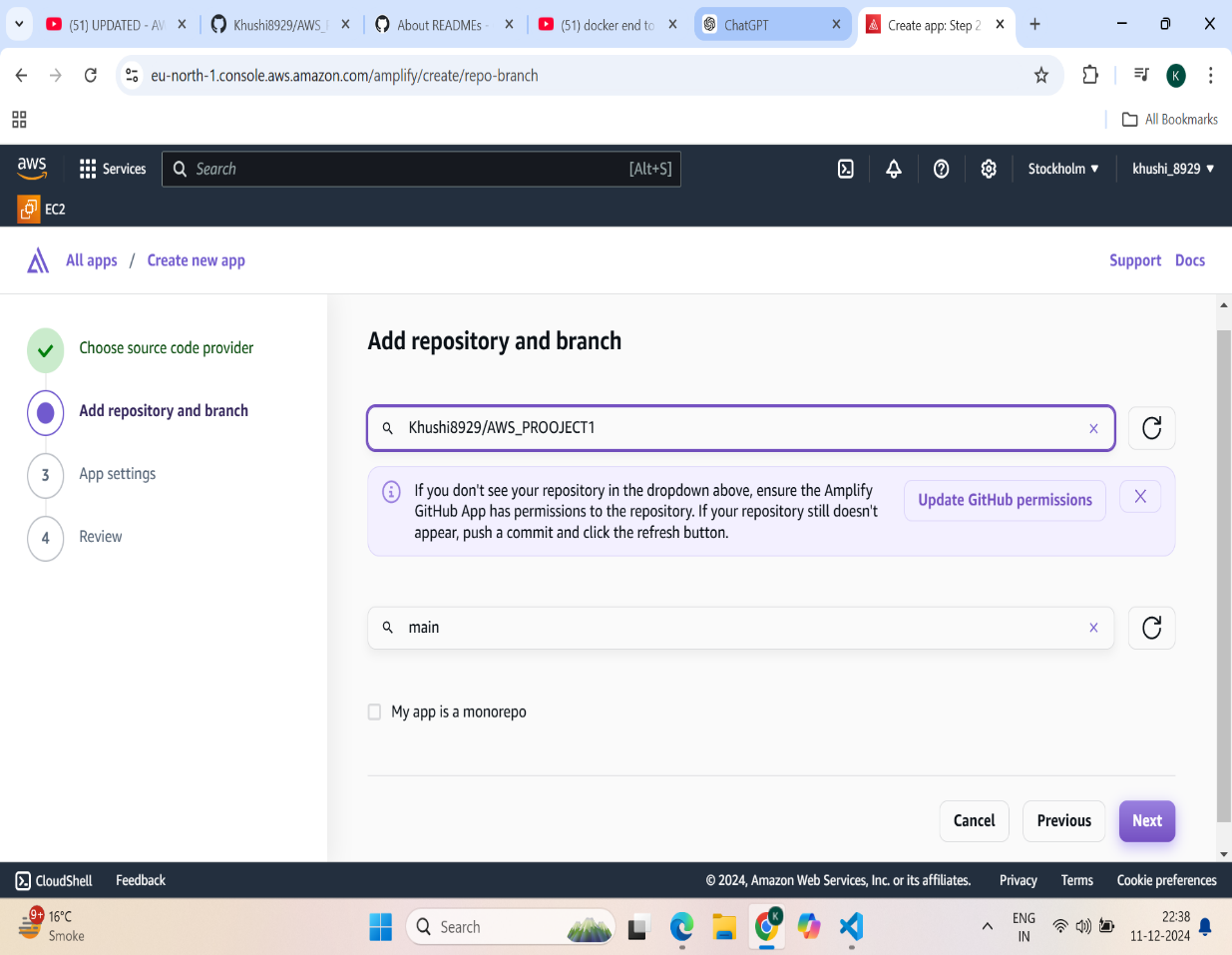
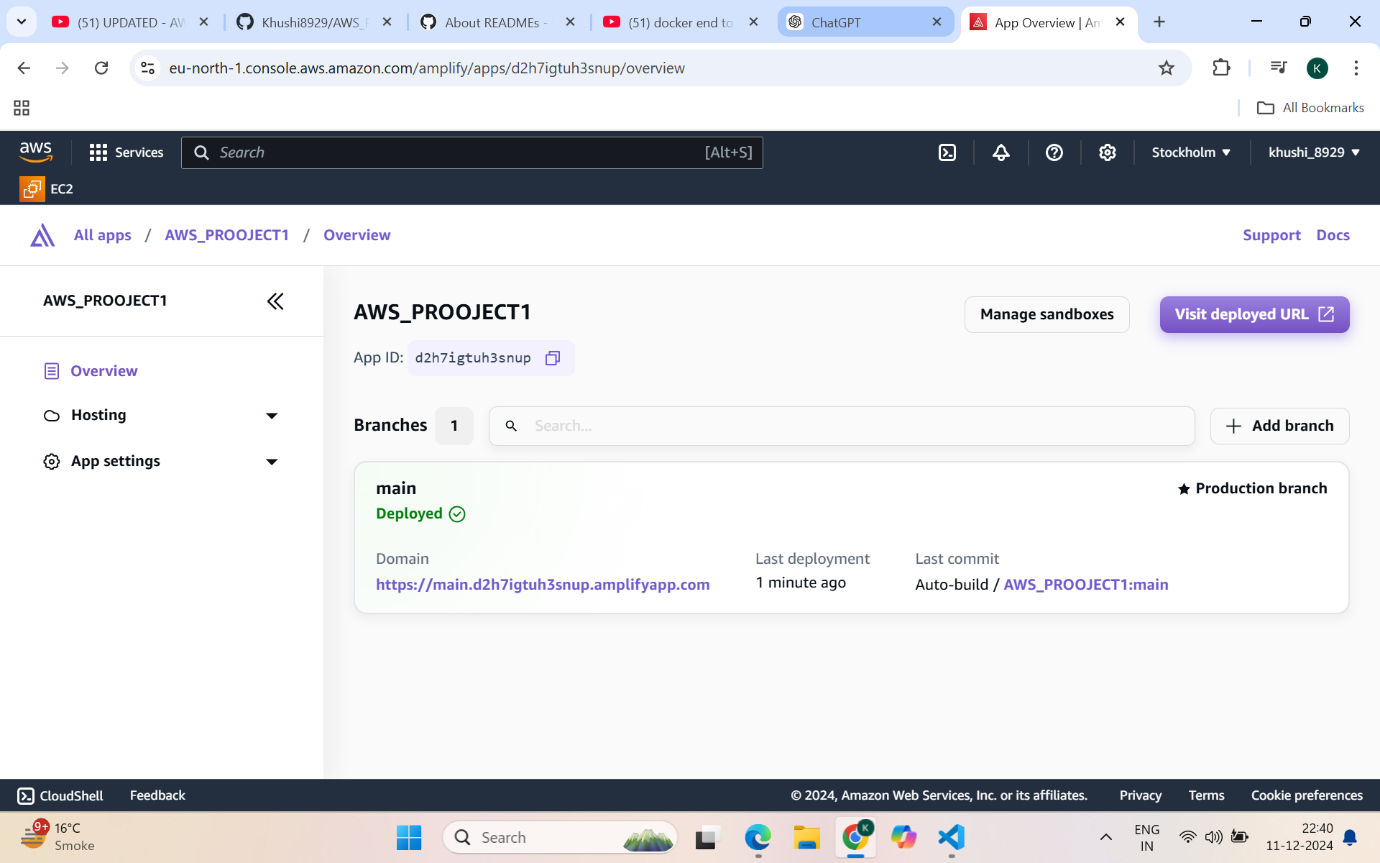
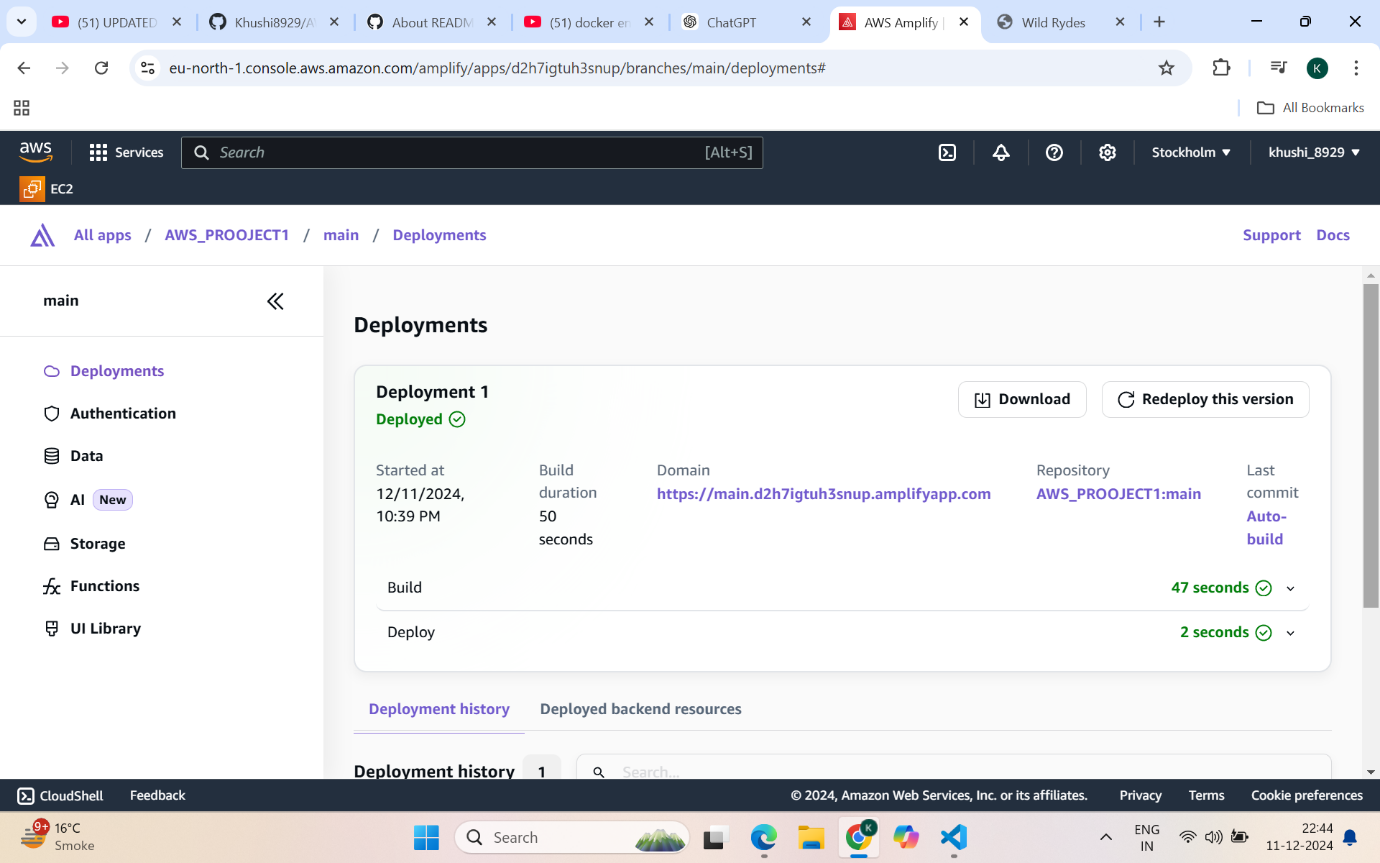
**Step1:** Creating a new GitHub repo using a template.

**Step2:** Creating a new app for hosting in AWS Amplify. Testing out the Amplify deployment by launching our website. Updating code in GitHub .

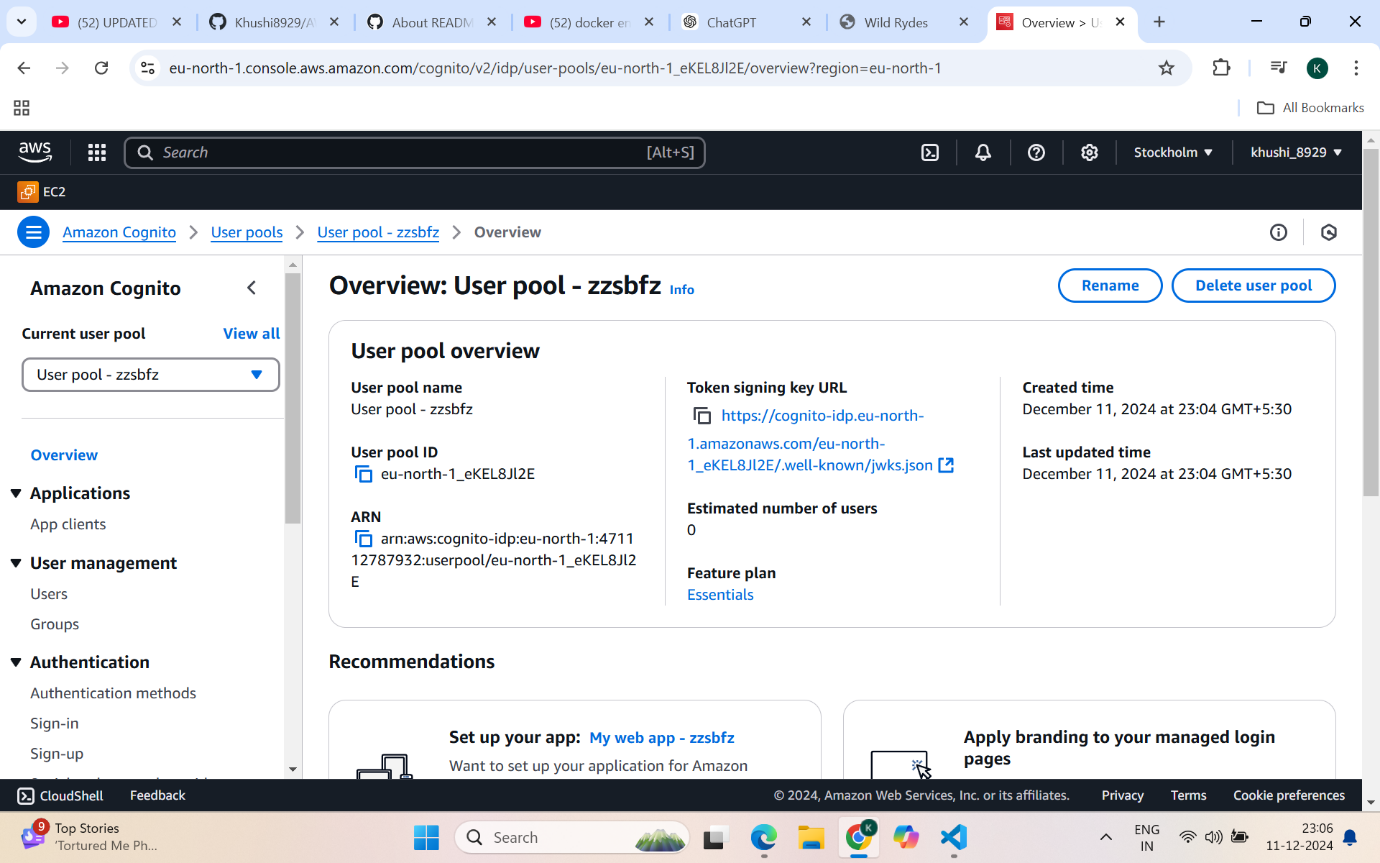


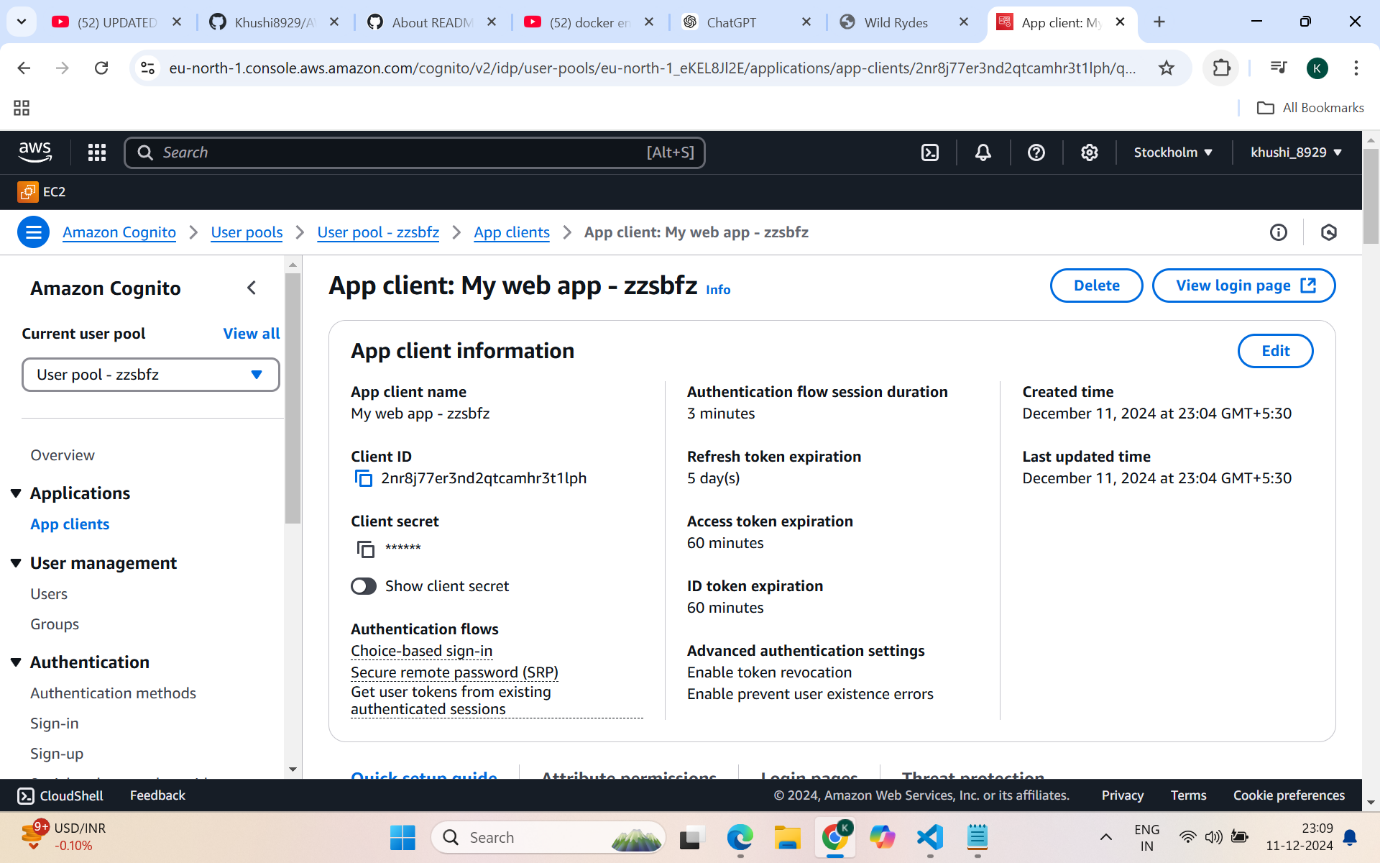


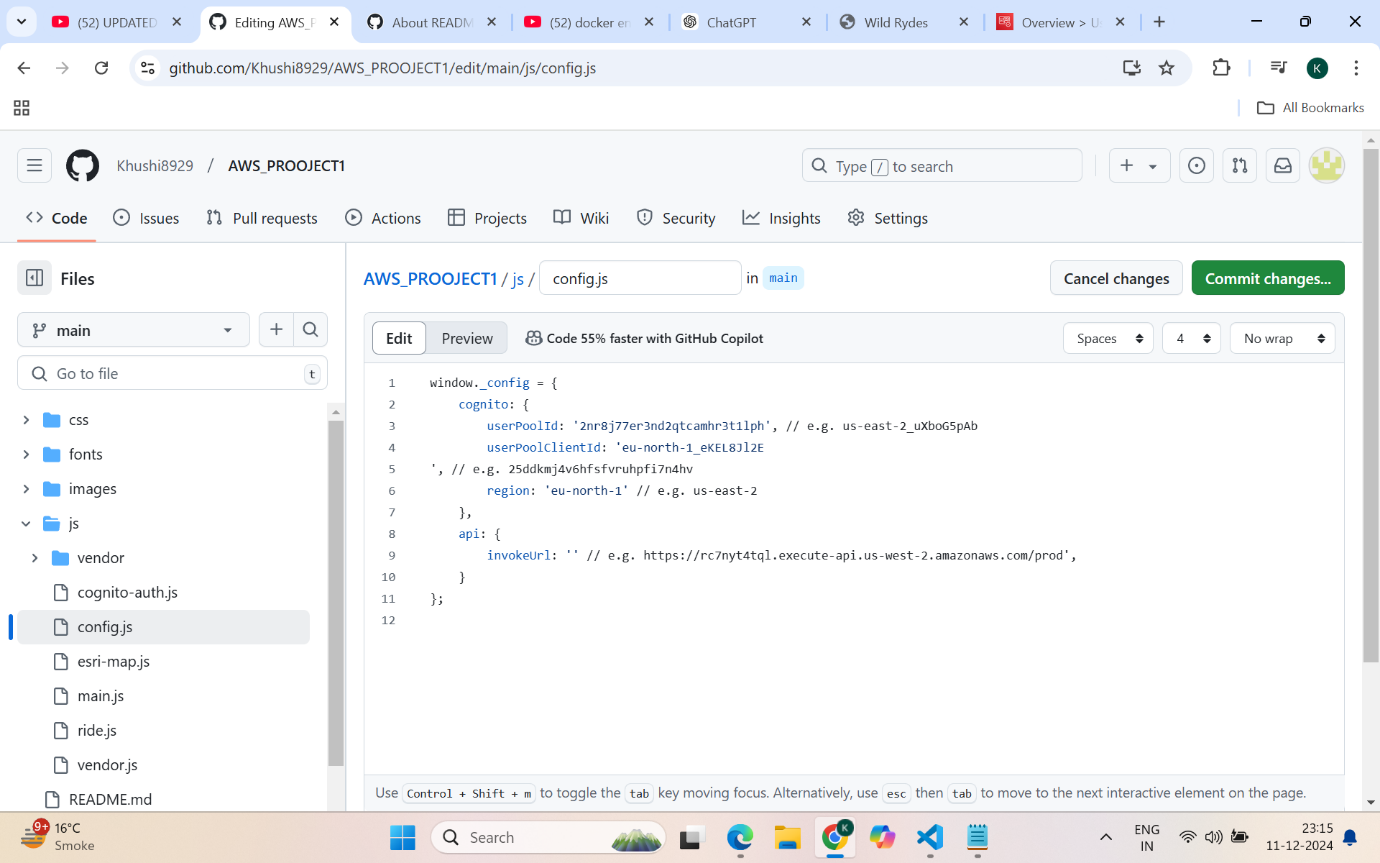


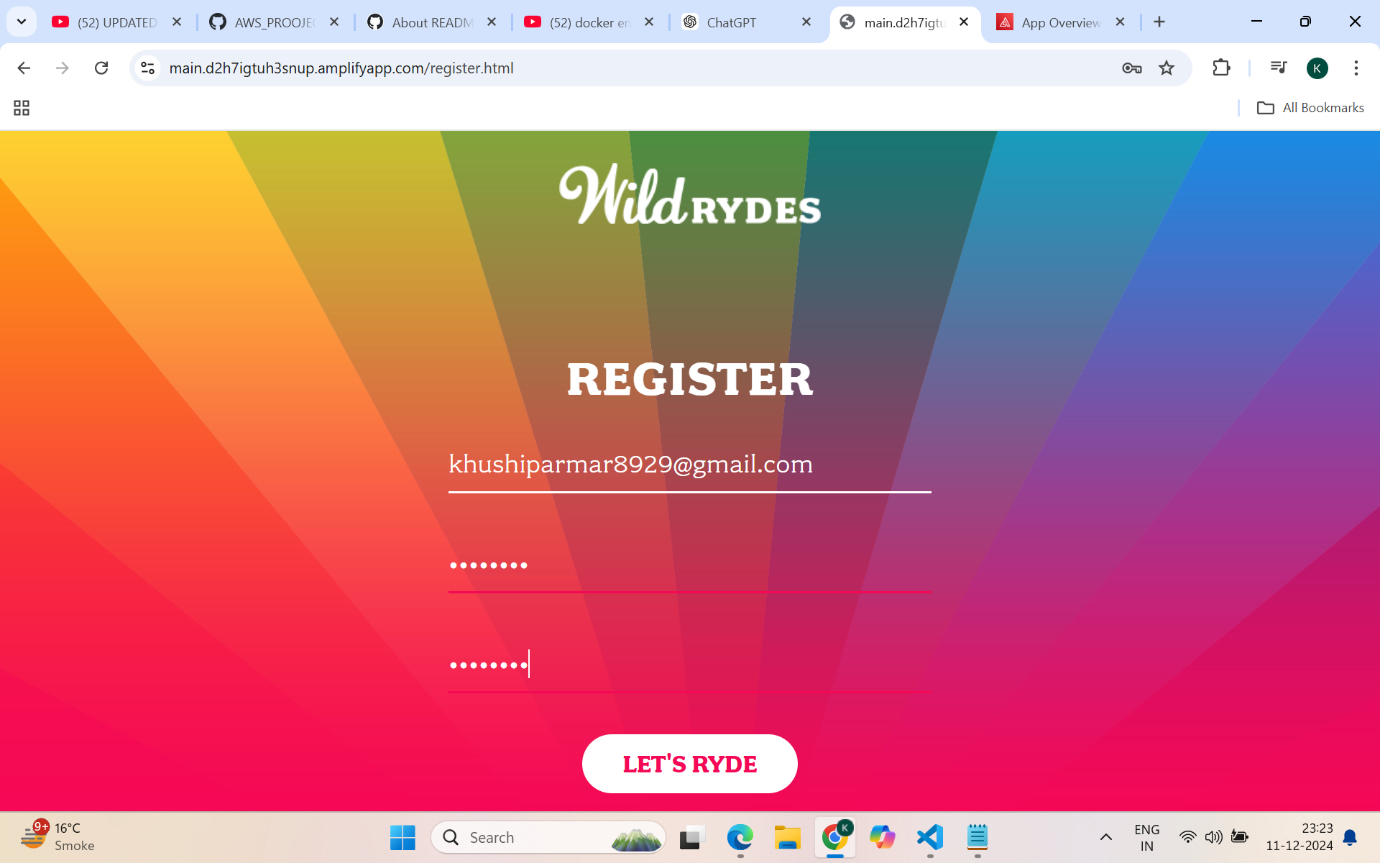


**Step3:** Setting up Amazon Cognito for user authentication. Creating a new user pool in Amazon Cognito. Updating the app configuration file to use the Amazon Cognito user pool Testing Cognito integration by doing user registration and login.

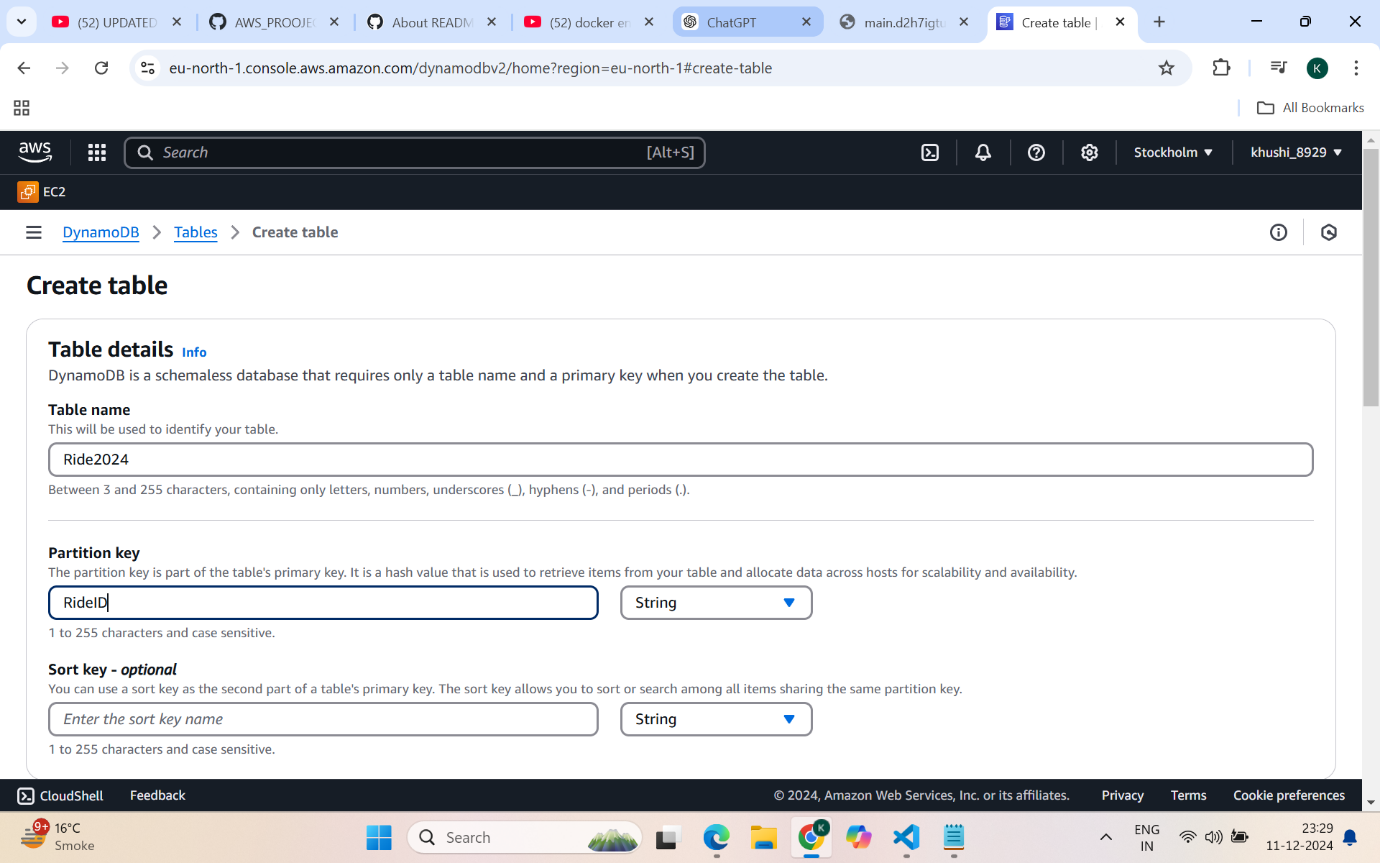




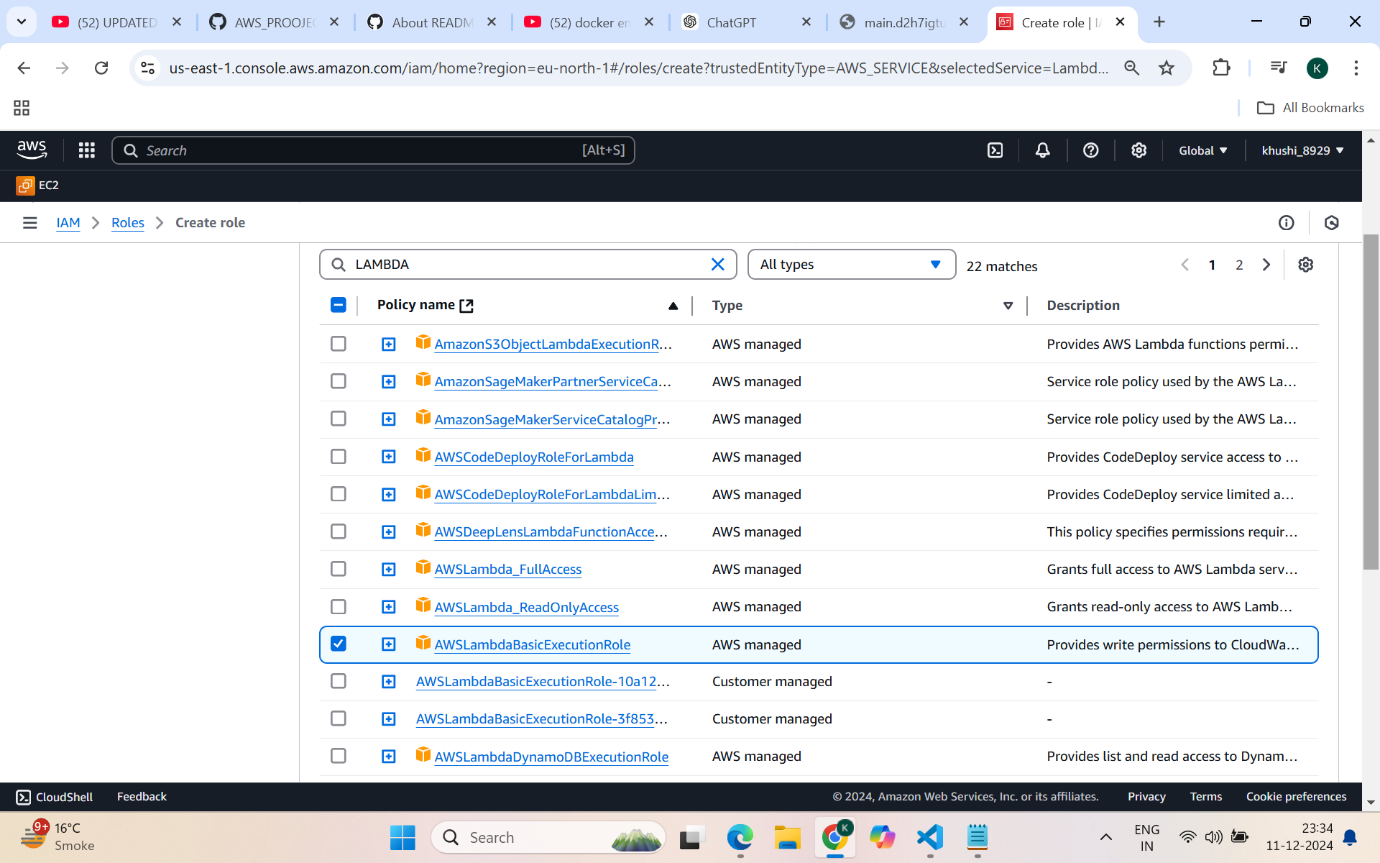




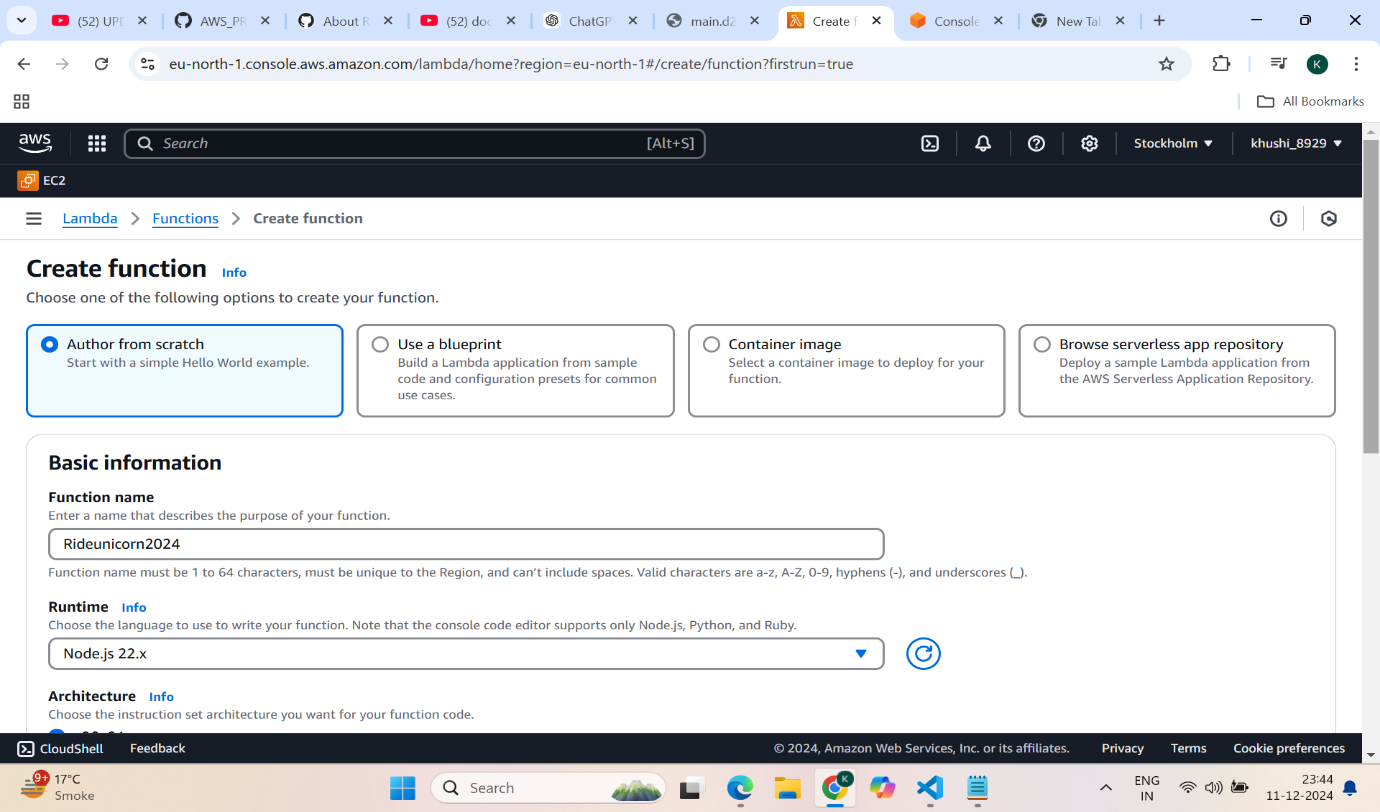
**Step4:** Implementing ride sharing functionality with Lambda and DynamoDB .Creating a new DynamoDB table

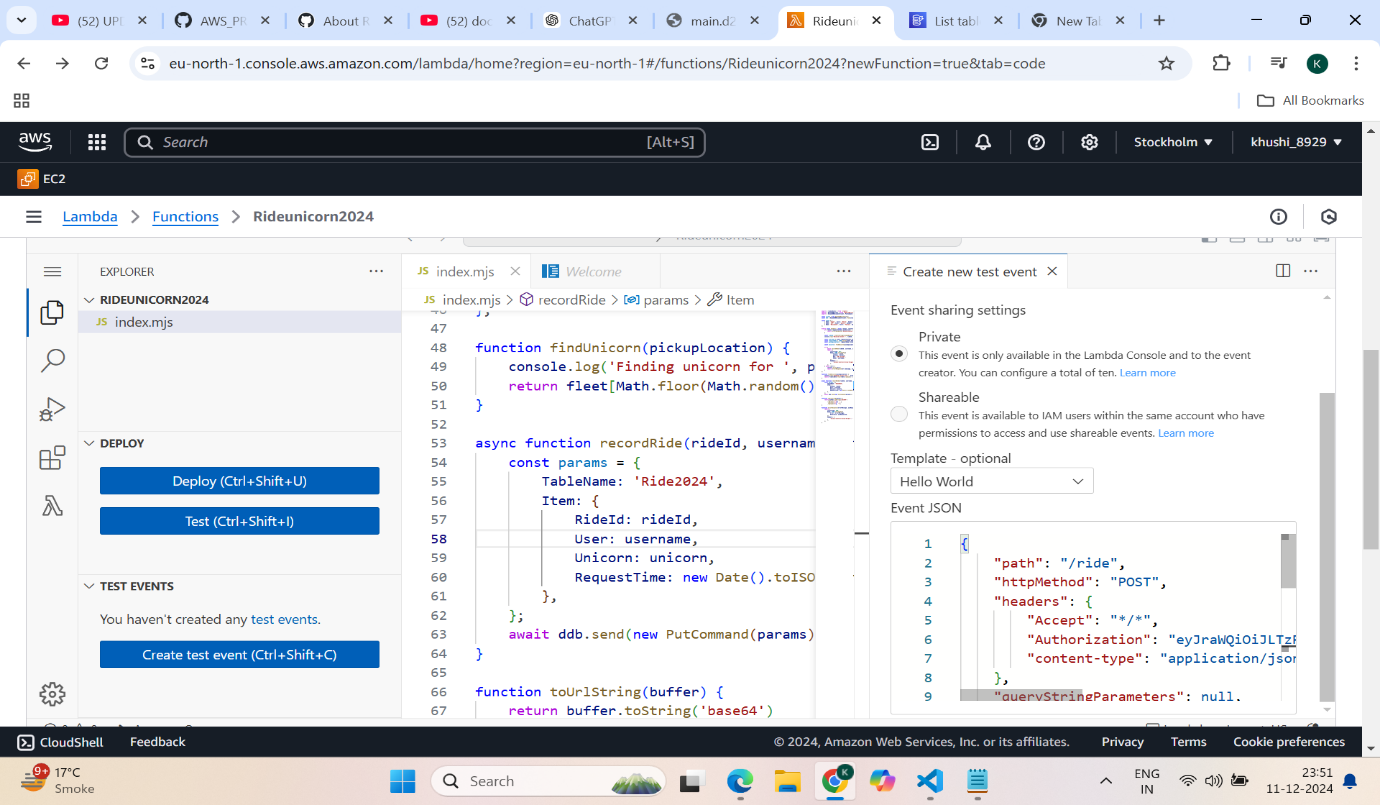


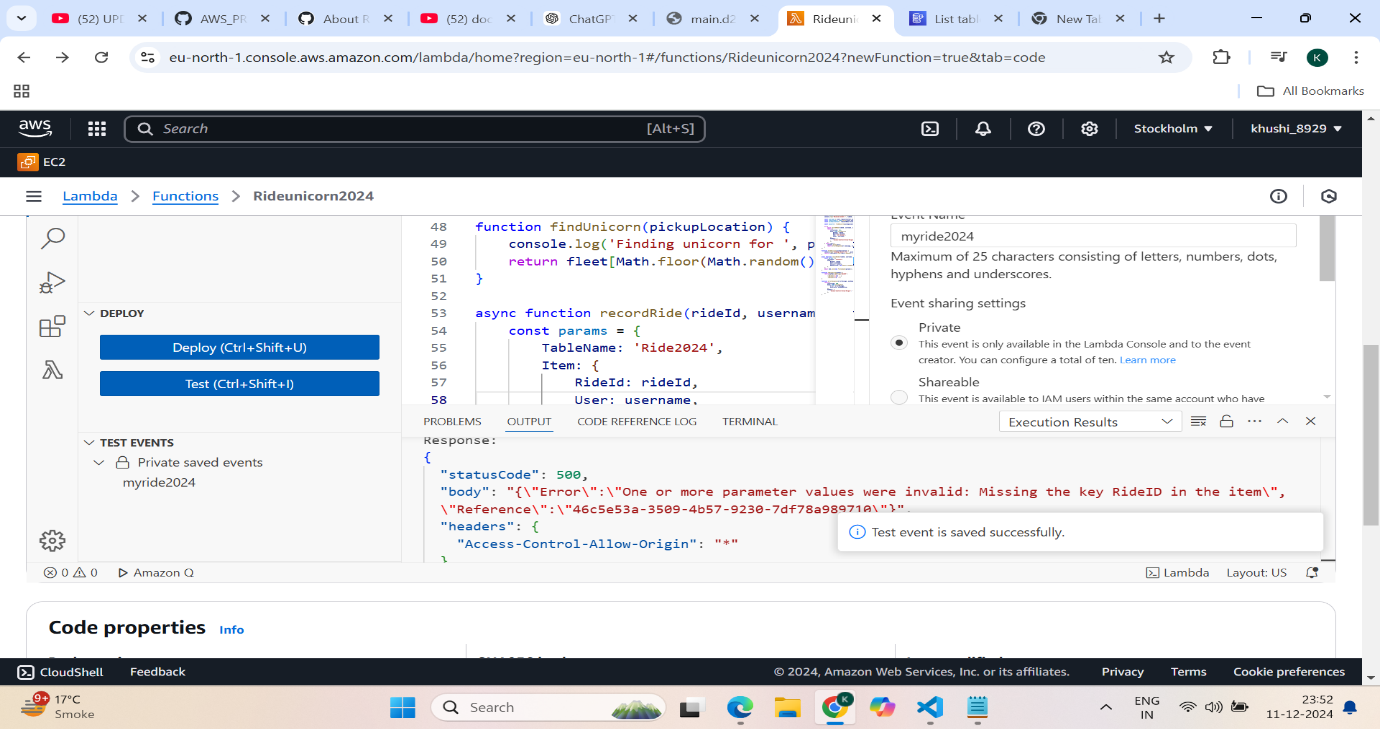
**Step5:** Creating an IAM role to be used for a Lambda execution role, allowing PutItem on DynamoDB table.



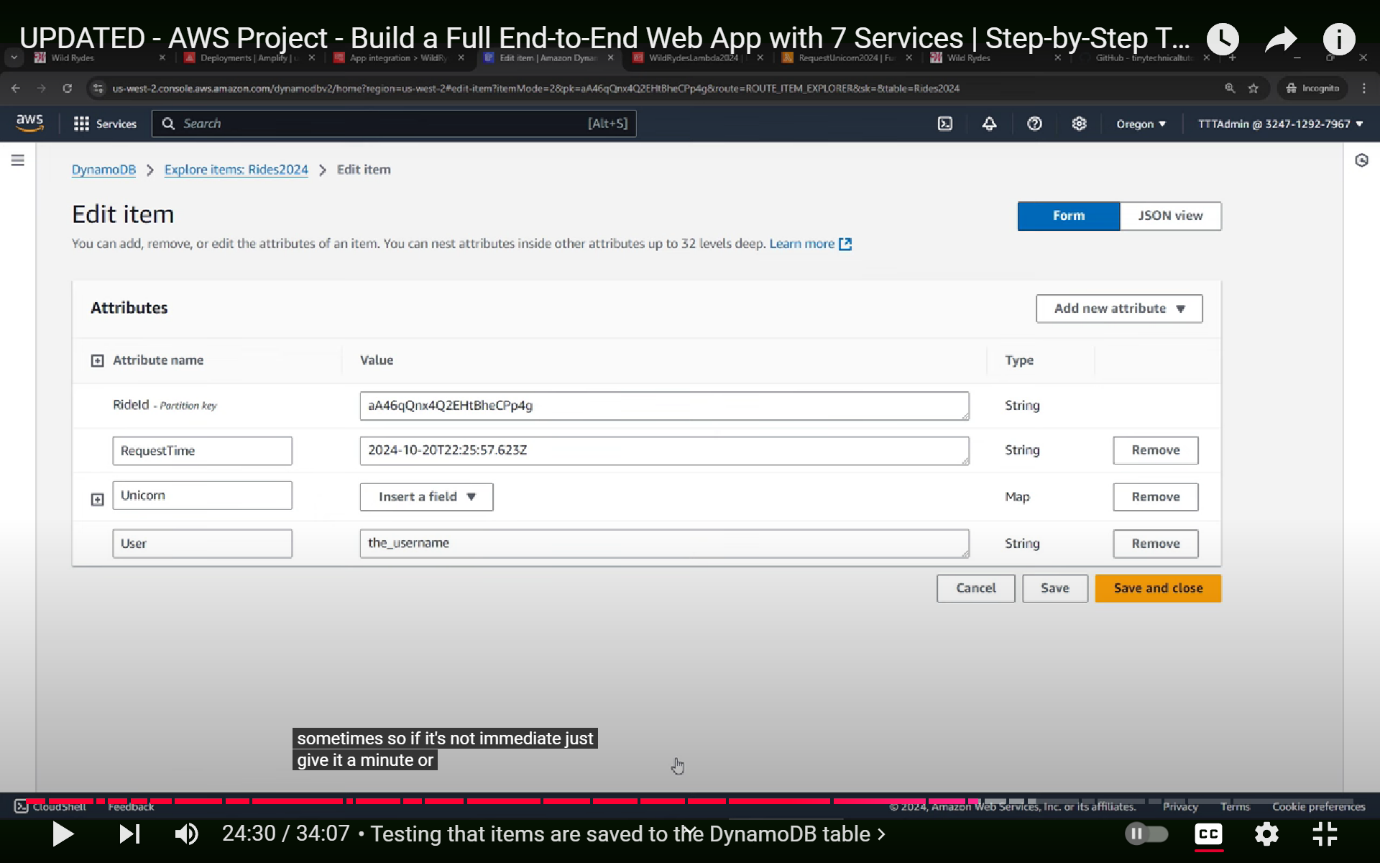
**Step6:** Creating a new Lambda function to choose a unicorn and write the ride sharing info to DynamoDB .Deploying Lambda code and executing a test event.



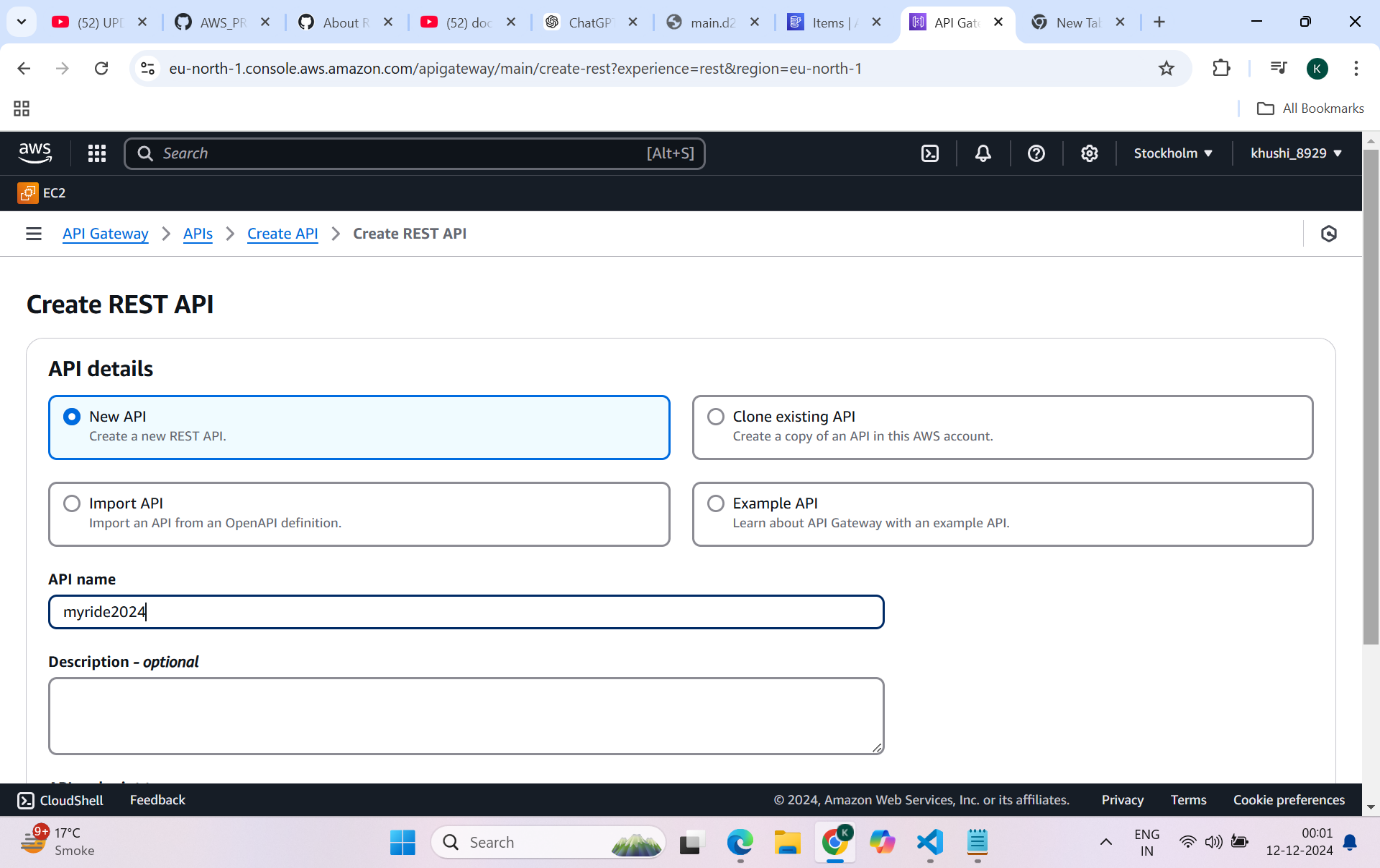


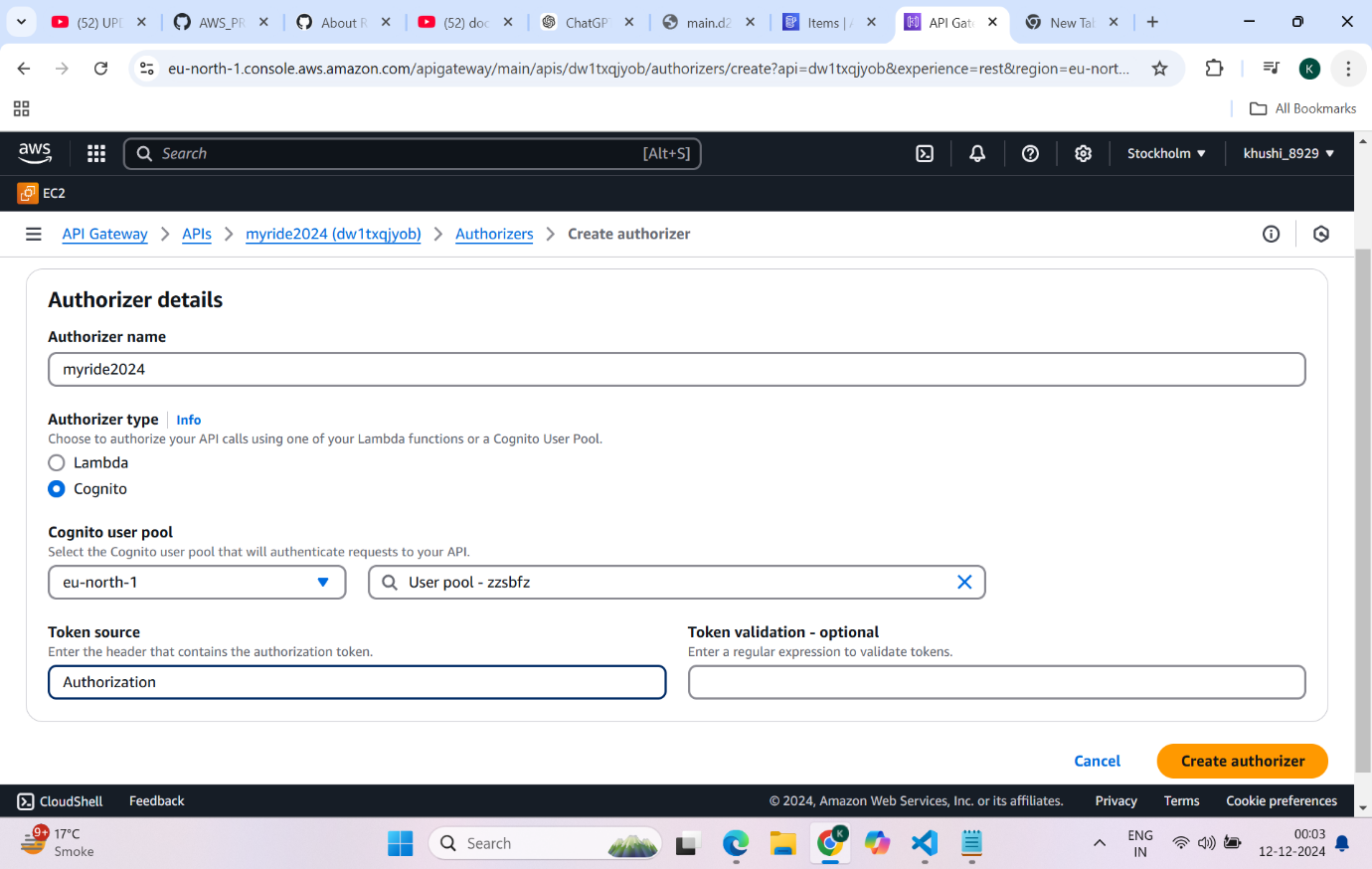


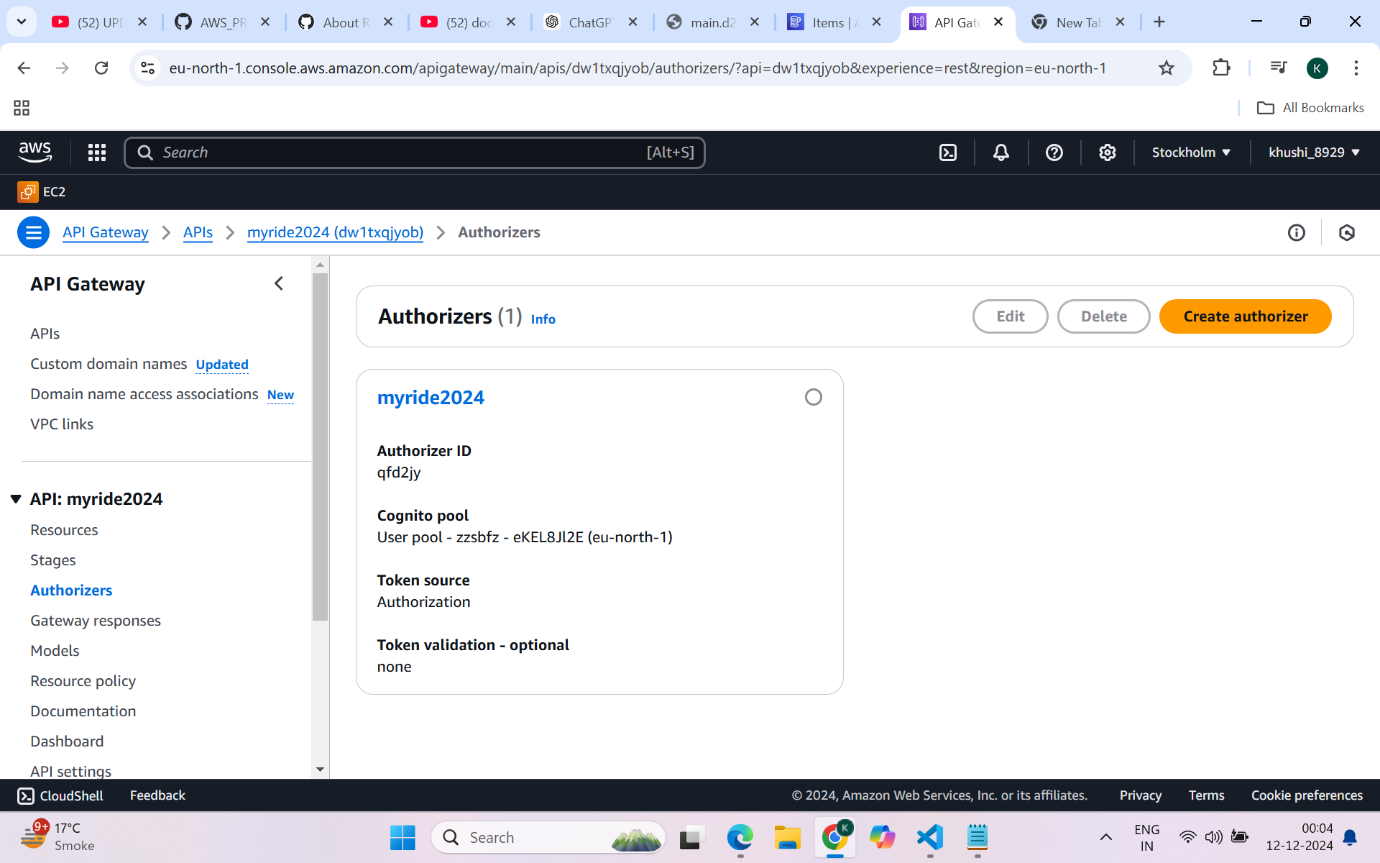
**Step7:** Testing that items are saved to the DynamoDB table



**Step 8:** Setting up API Gateway to invoke the ride sharing functionality. Creating a new REST API in API Gateway to invoke a Lambda function.Creating an authorizer so API Gateway can work with Cognito.Creating a resource and POST method in API Gateway for Lambda integration.Enabling CORS (Cross Origin Resource Sharing) for the API Gateway

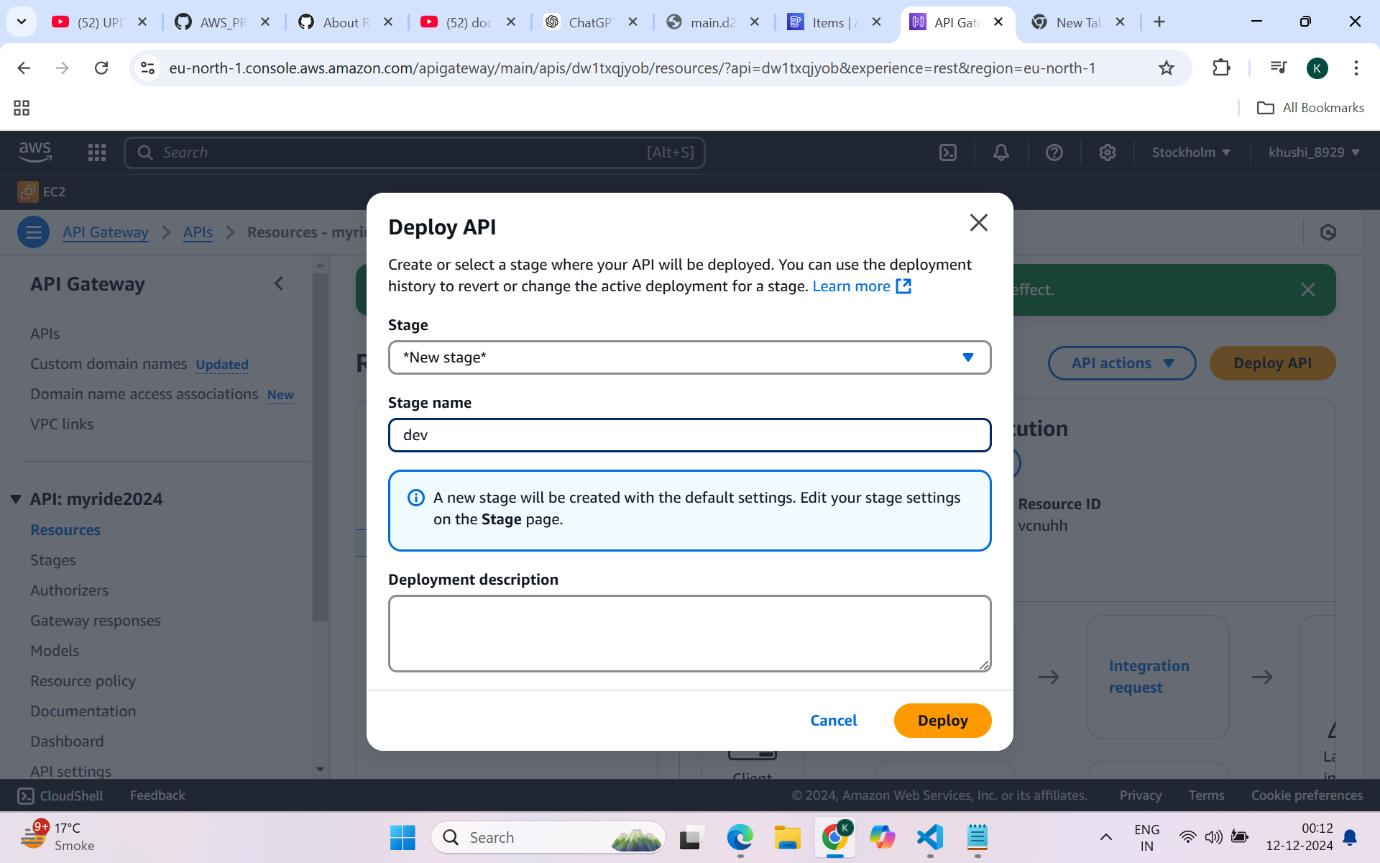


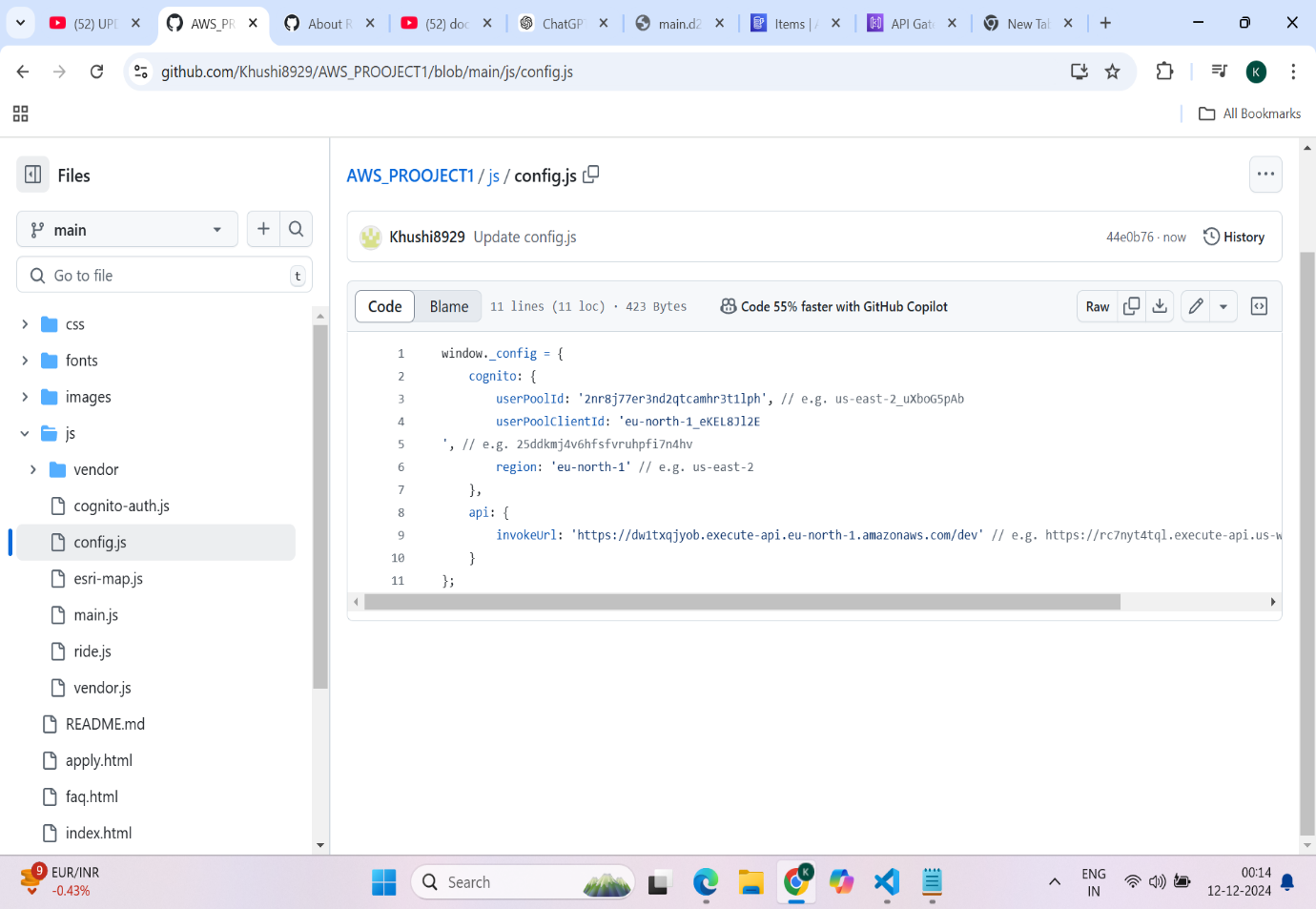


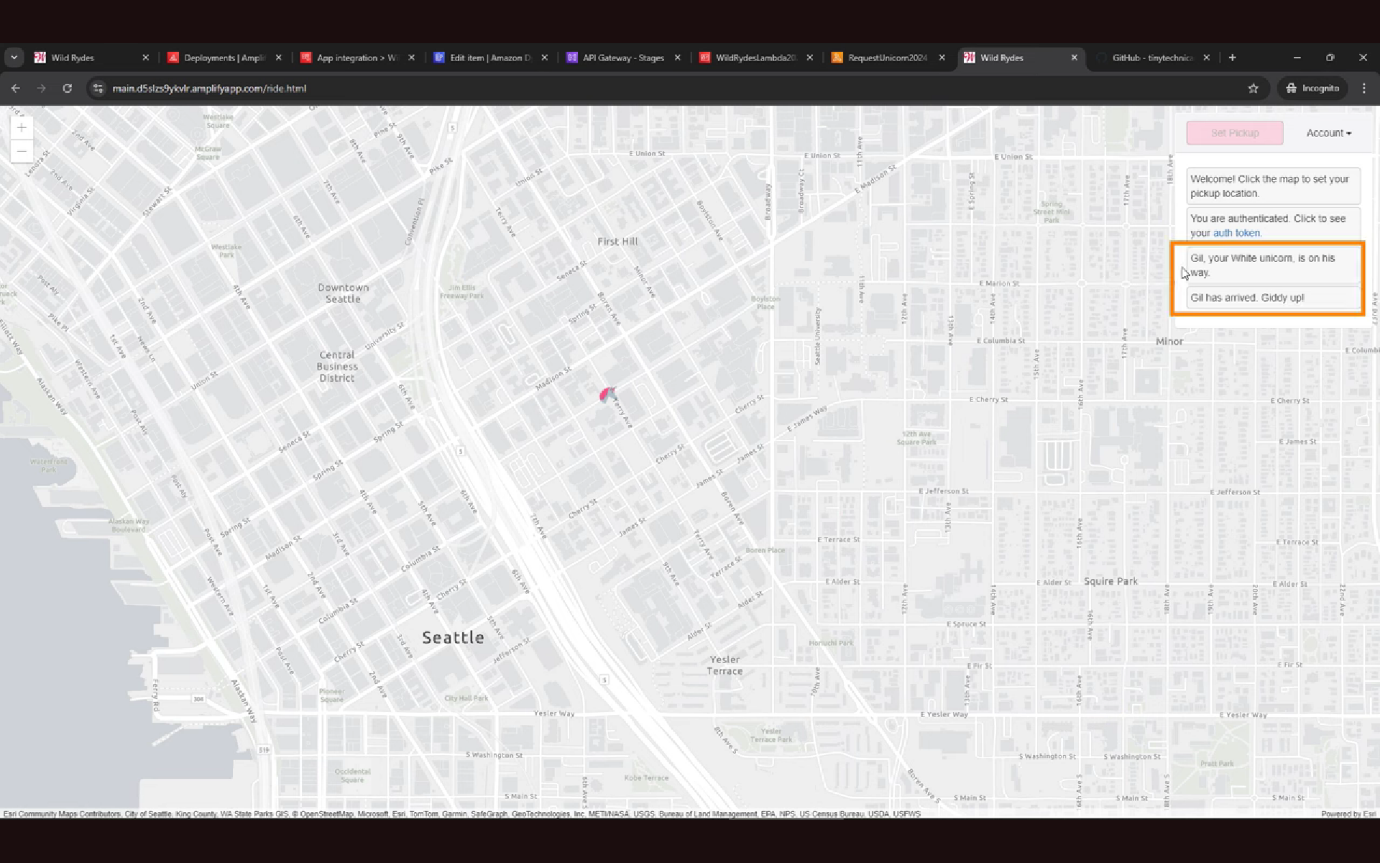


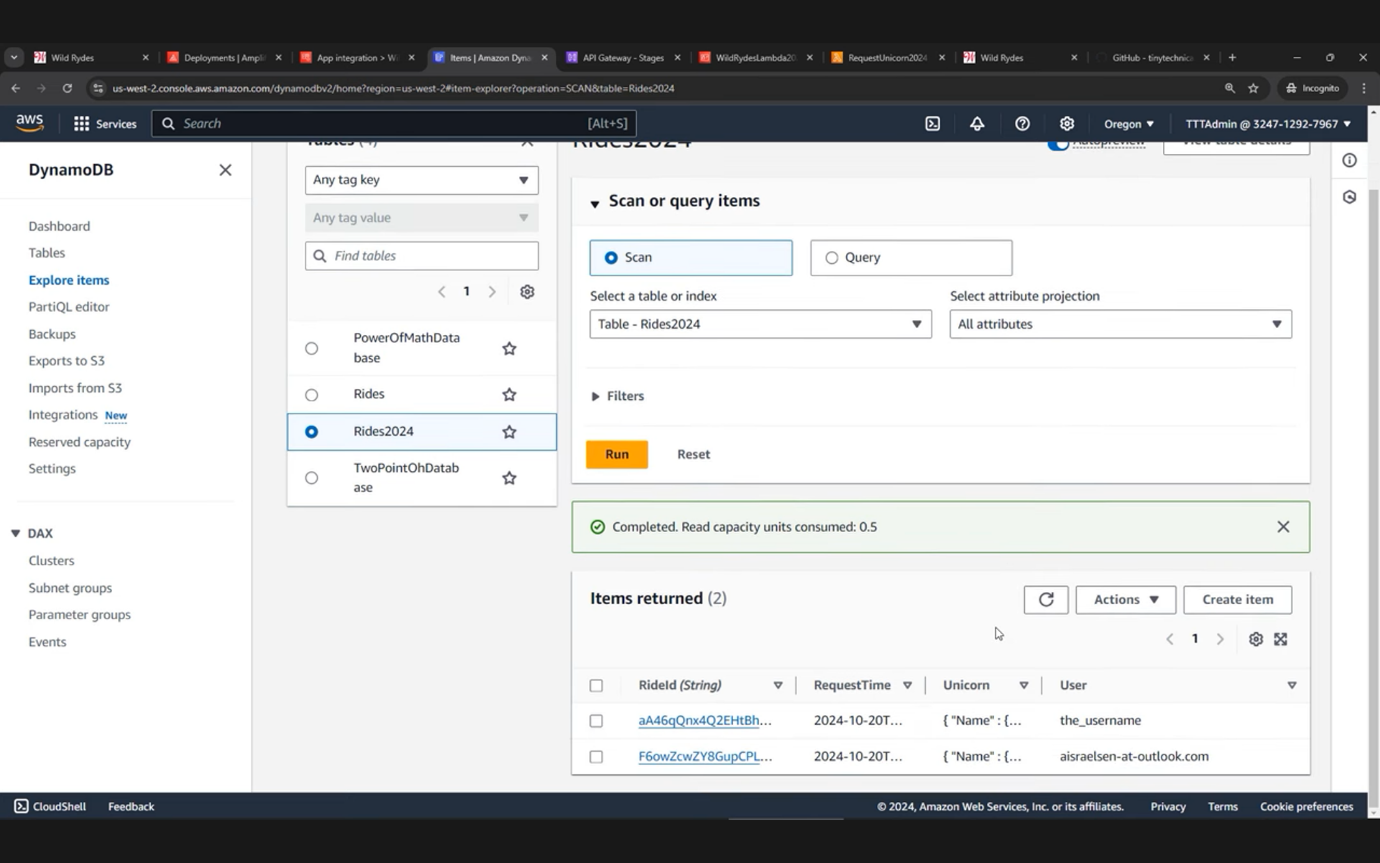


**Step 9:** Deploying the API from API Gateway .Updating the config file for the new Invoke URL from API Gateway .Testing our final application.









**ARCHITECTURE:**

