ENGR-E 534 BIG DATA APPLICATIONS
Assignment 8
Shalini Kothuru (University ID 2001096463)

Note : I have attached dataset, notebook file - html and ipynb, index.html file and lambda function – python file. Index.html file and Lambda function varies from the screenshot shown in this document as I have updated while working on web app but forgot to take sreenshot.

Architecture Design:

Data source: I have downloaded data from **Kaggle.** Dataset has 1025 records with 14 columns including target variable(0 = no disease and 1 = disease) for heart disease.
Data storage: I have uploaded the dataset to **S3**. S3 also store the models once trained.
Data preparation: I have created **notebook instance in sagemaker** to explore, preprocess, prepare data and train model.
Model building and training: Sagemaker has built-in algorithms. I have used **xgboost and linear learner built-in model** for my binary classification problem. The sagemaker training jobs use the dataset stored in S3 to train the model and the trained model back in S3 again.
Model deployment: the trained models both xgboost and linear learner are deployed to **sagmaker endpoints**. These deployed models will be used by lambda function to predict the target variable
API and application layer: AWS Lambda hosts the backend code in python file which makes call to sagemaker endpoints. It processes the incoming requests from app and sends them to model in sagemaker. **API gateway** is used to trigger lambda function on http requests.
Frontend web application: S3 is used to the index.html file.

Workflow:

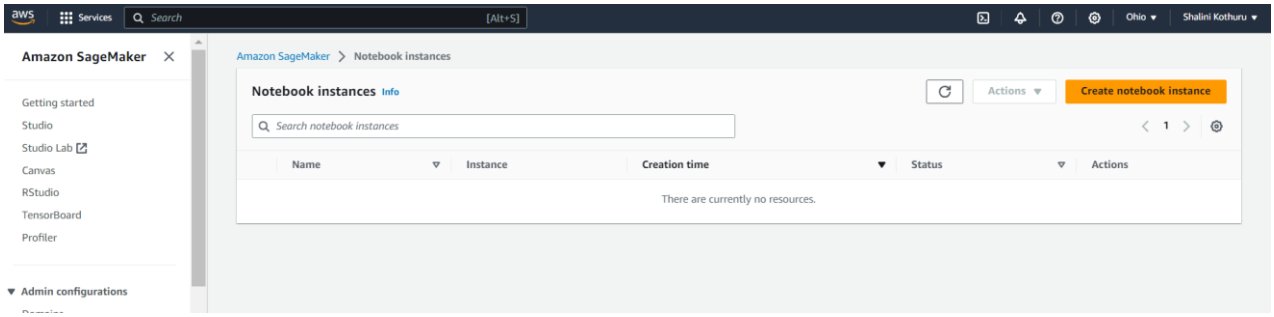
The user enters details basically feature values in web application hosted on S3. The web application sends a post request to API gateway with the details entered which routes the request to lambda function. Now lambda function processes the request and invokes sagemaker endpoint and sends the prediction result back to API gateway which in turn returns to web app. The web application displays the result.

Configuration Details:

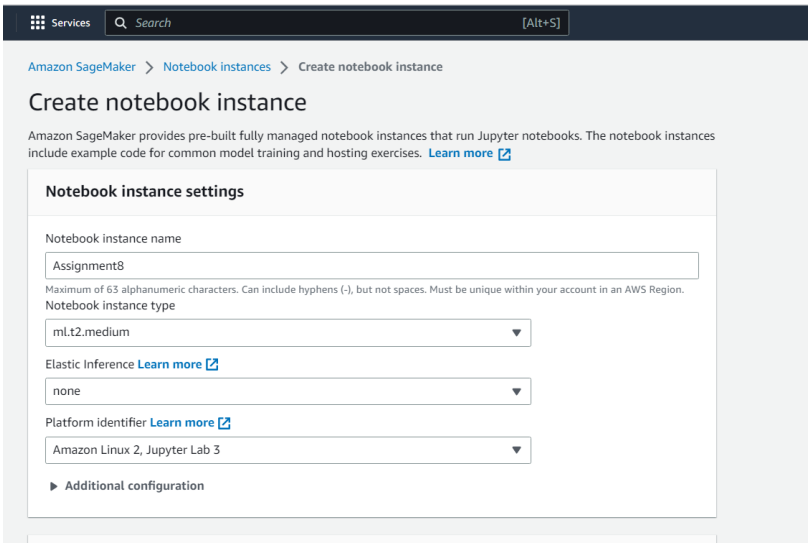
I have explained configuration details step by step using screenshots.

AWS Sagemaker for ML project:

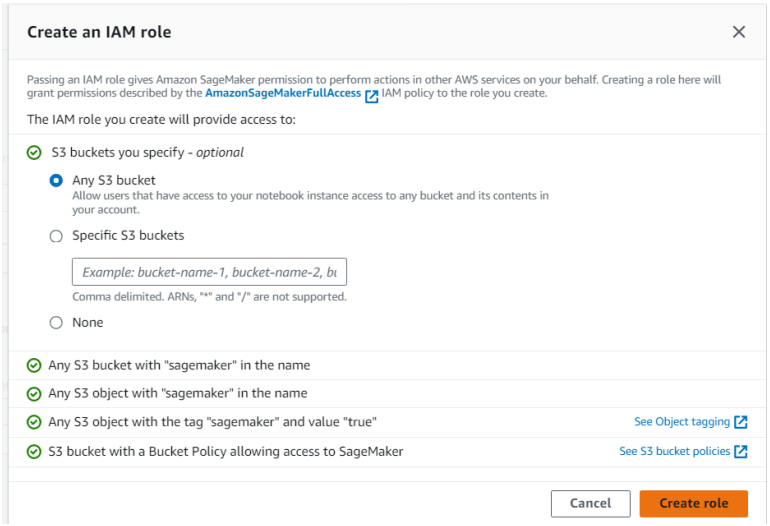
Login in to AWS management console and search for sagemaker. Open Sagemaker service.
In left side menu, you will find notebook, click on notebook instance under notebook which takes you to below screen



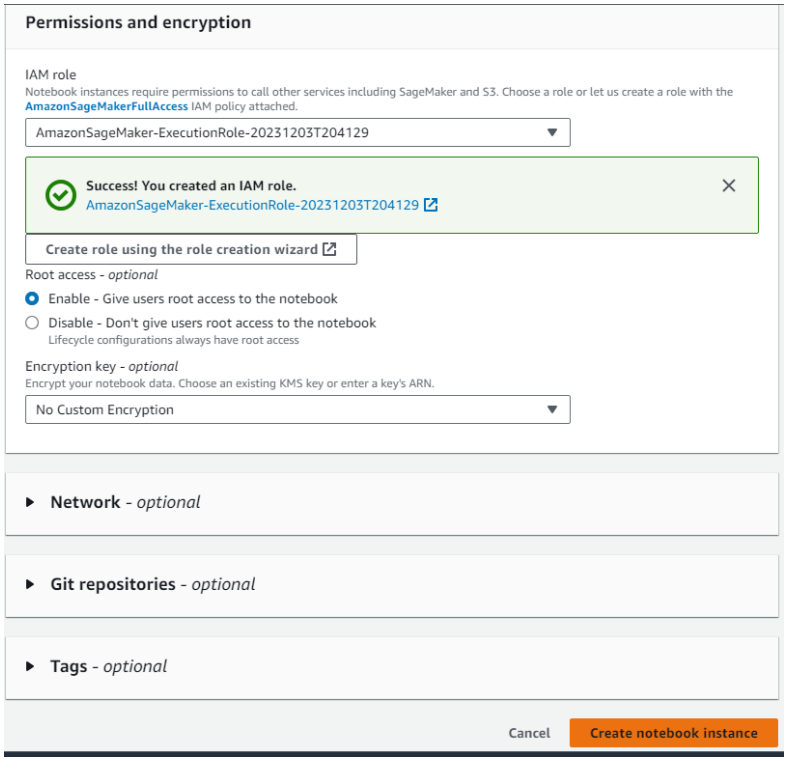
Click on create notebook instance. Enter details as below



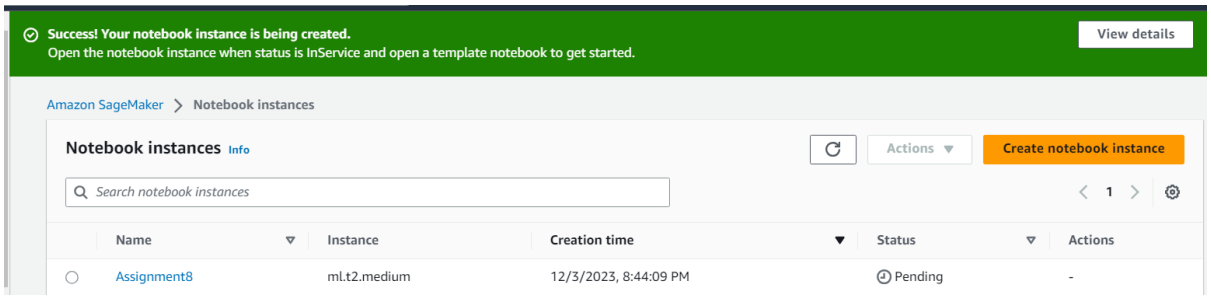
Next, click on create role.



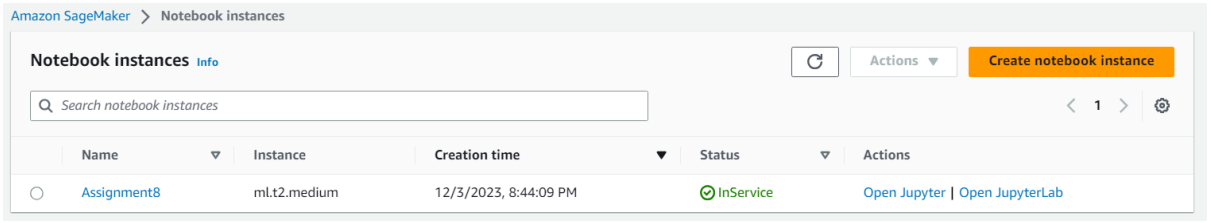
Next click on create notebook instance.



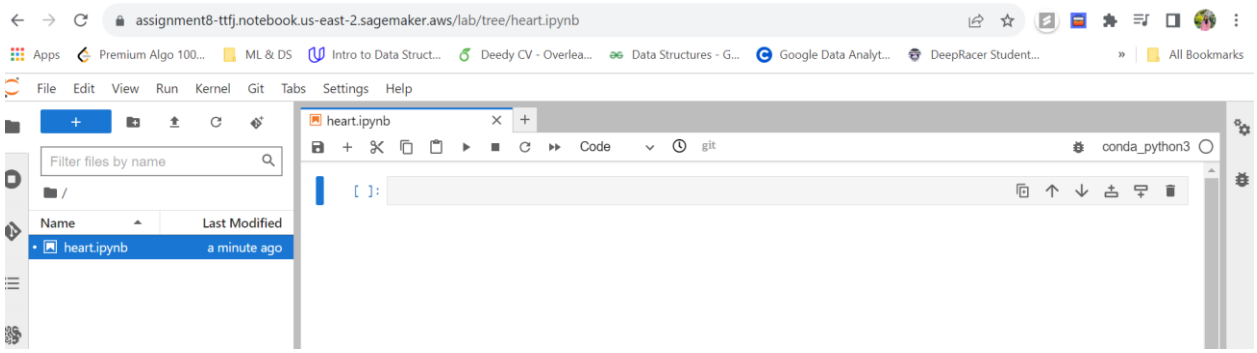
Notebook instance is created successfully



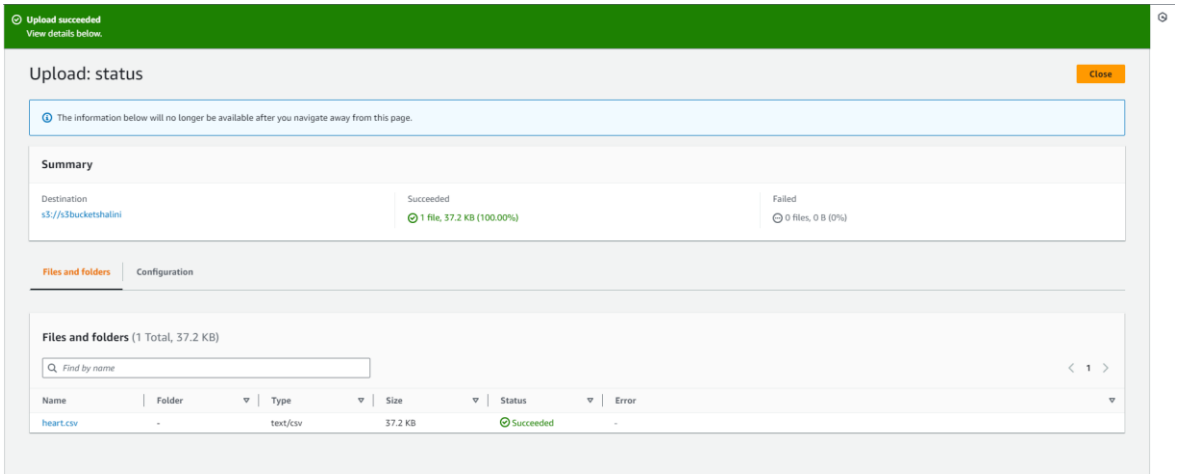
One status is in service, click on open jupyter/jupyterlab. I opened Jupyter lab



On clicking it, it will redirect to notebook UI. Select kernel as Python. I renamed my file to heart.ipynb



I have uploaded the heart.csv file to S3 bucket



Notebook:

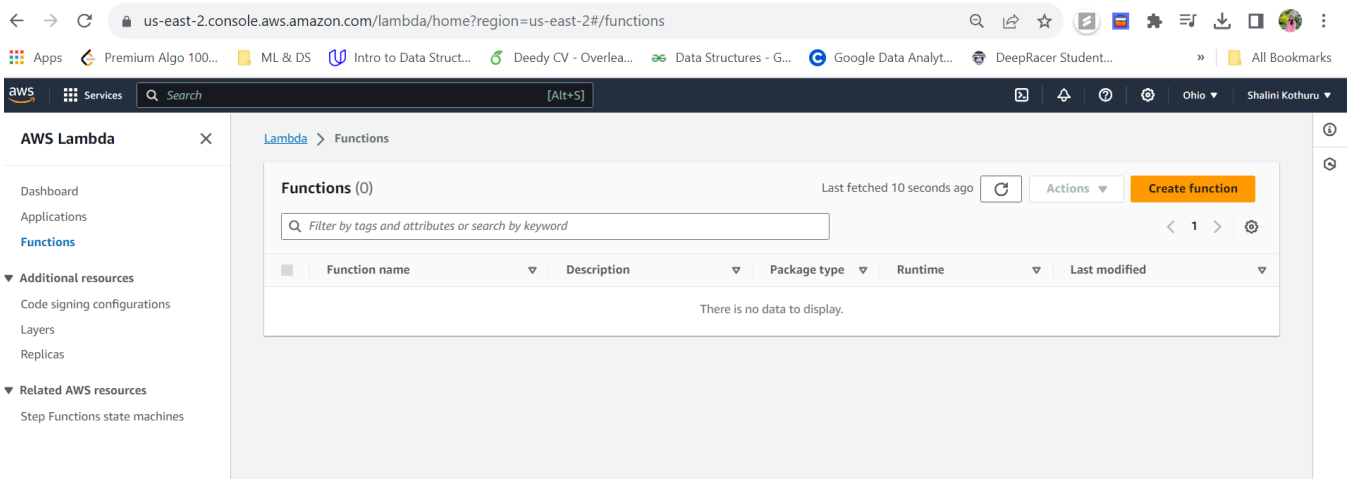
I have uploaded the notebook in canvas where I did data exploration, processing, feature selection, building, training and deploying the model.

Results :

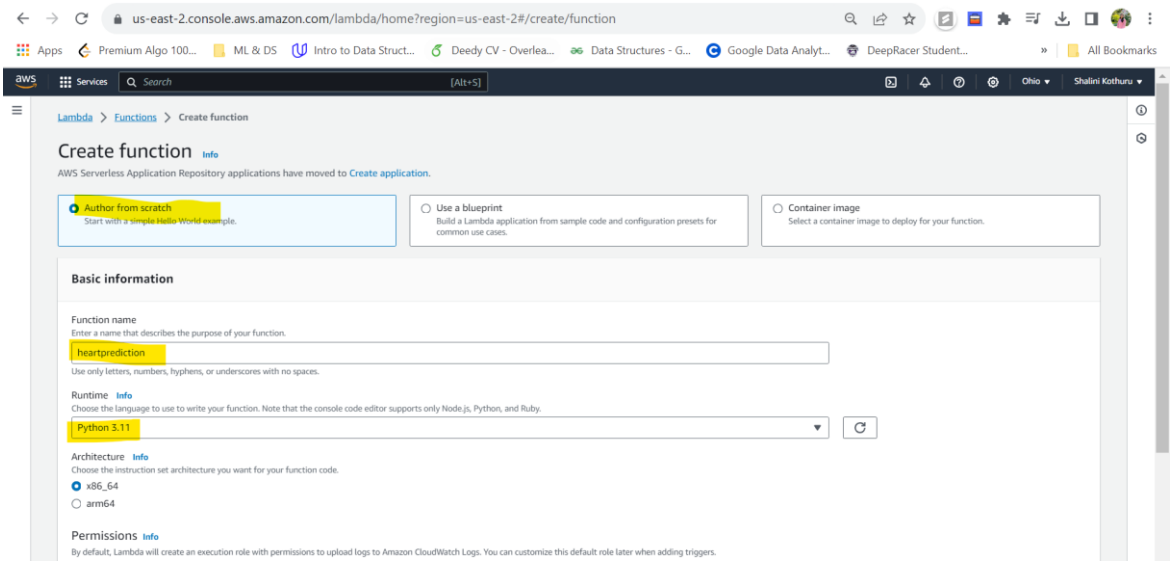
My XGBoost model resulted in 85.7% while Linear learner model resulted in 81.4 accuracy. Since XGBoost model performed better than linear learner model, I used XGBoost model to predict results based on inputs from user. I deleted enpoints after taking all the screenshots. Below is the process to create web app that takes inputs from user and predict the target variable indicating presence of heart disease.

Web application:

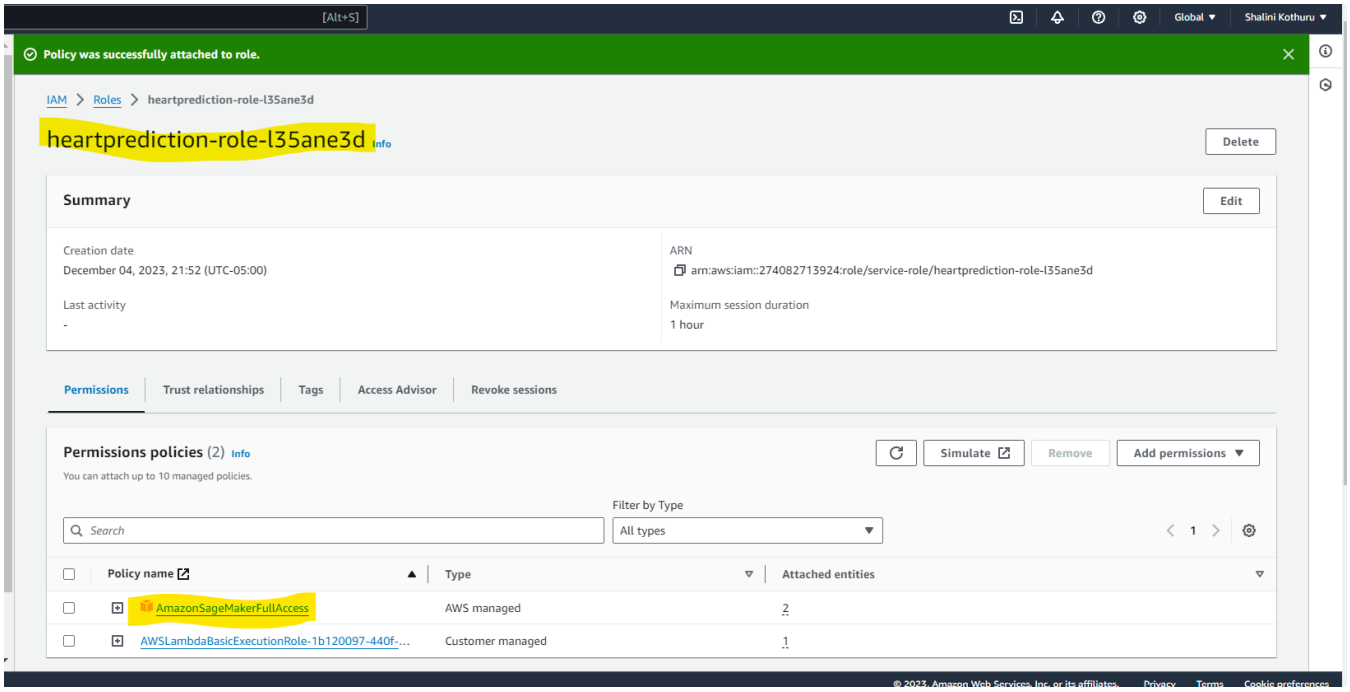
Go to AWS Management console and search for lambda and then click on create function



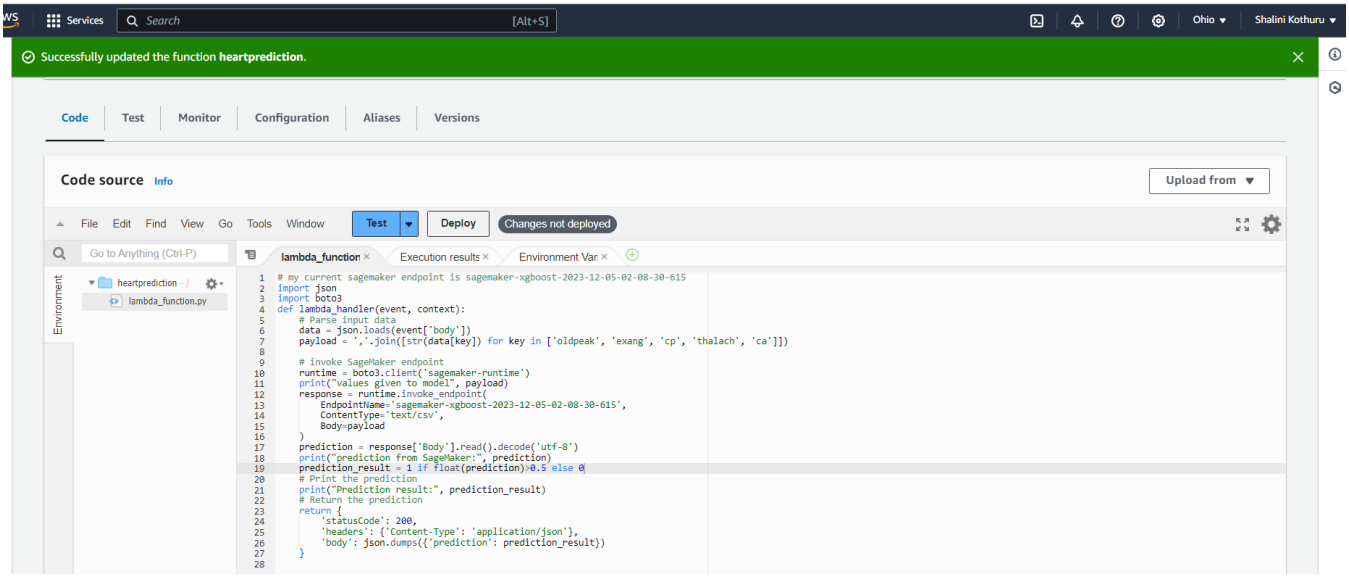
Add details as below and click on create function at bottom right corner of web page which creates function



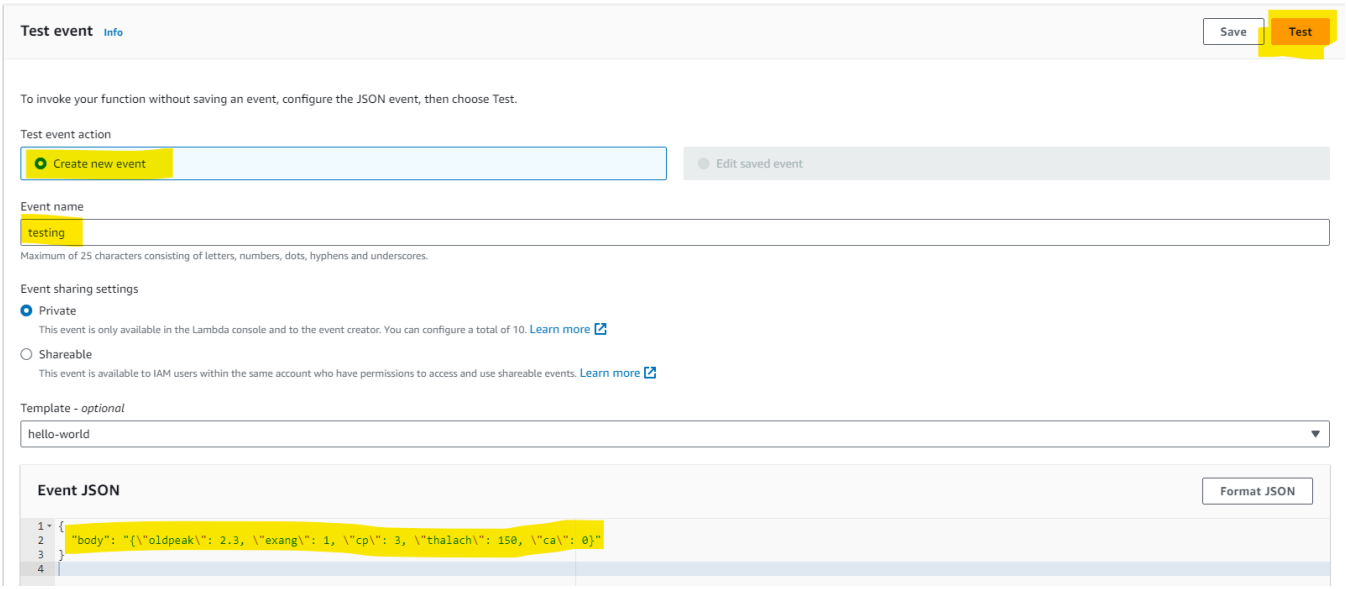
The function is created successfully. There is default role created when this function was created. I went to IAM services to update policy for that role. I have attached SAGEMAKERFULLACCESS for that role.



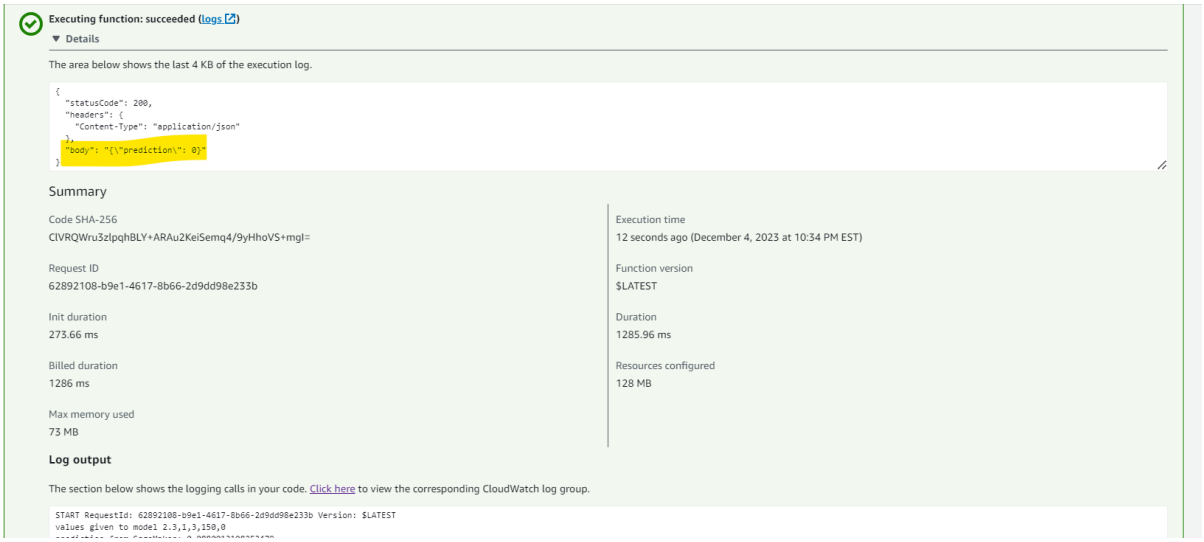
Now add code as below in code tab. My xgb performed better so using xgb endpoint for prediction.



Now click on test next to code tab and create a test event. I gave random values. Now click on test

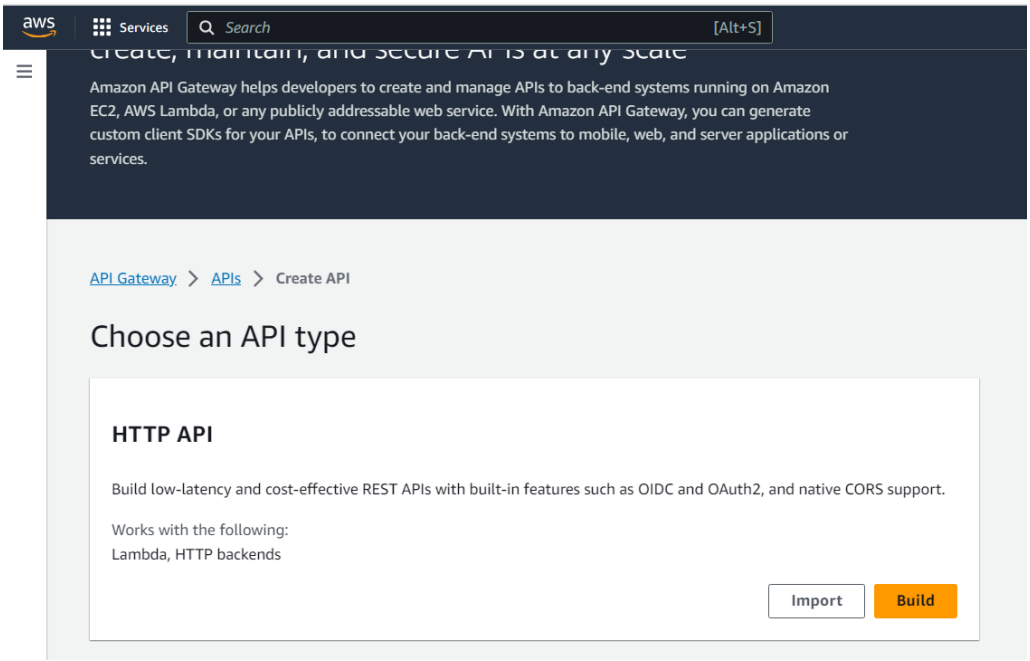


You can see logs in same page. It predicted 0 for this values

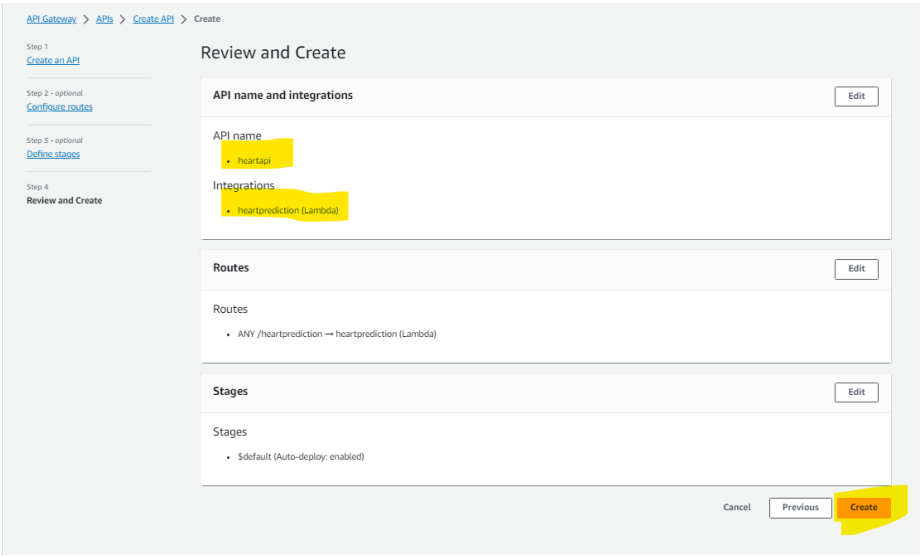


So my lambda function is working properly.

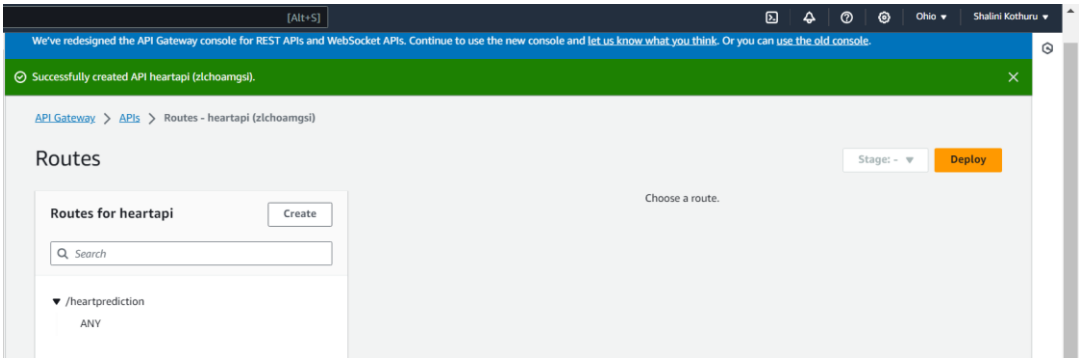
Now go to API gateway in AWS management console and click on HTTP API build



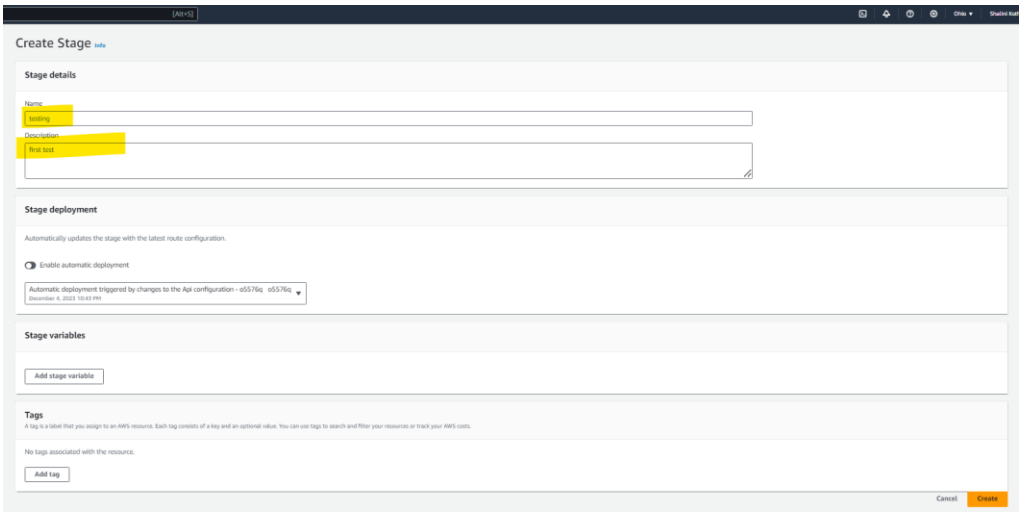
Now enter details as below and create API. You need choose lamda in step 1 and choose lambda function that you have created above.



Click on deploy API

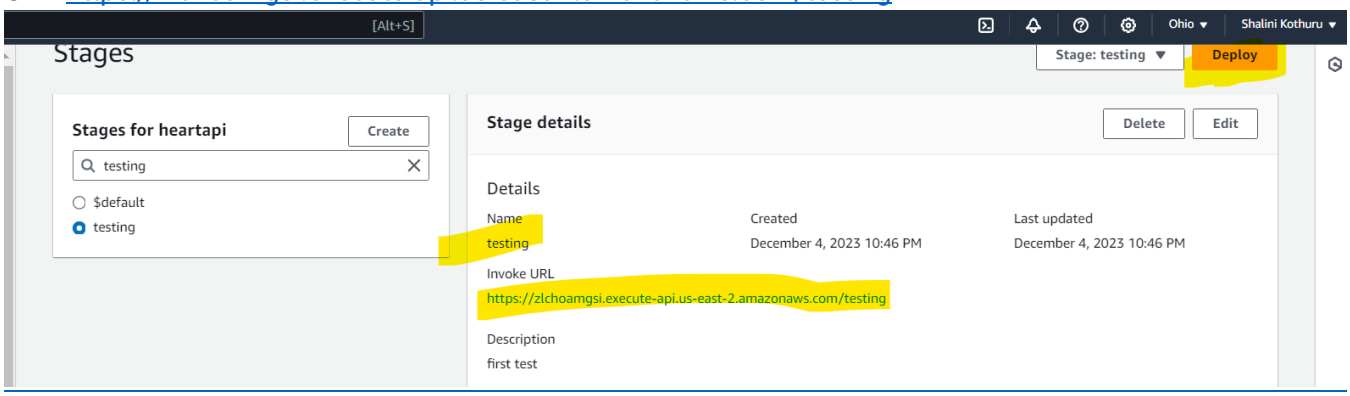


Once you click deploy, it asks for stage. I created stage

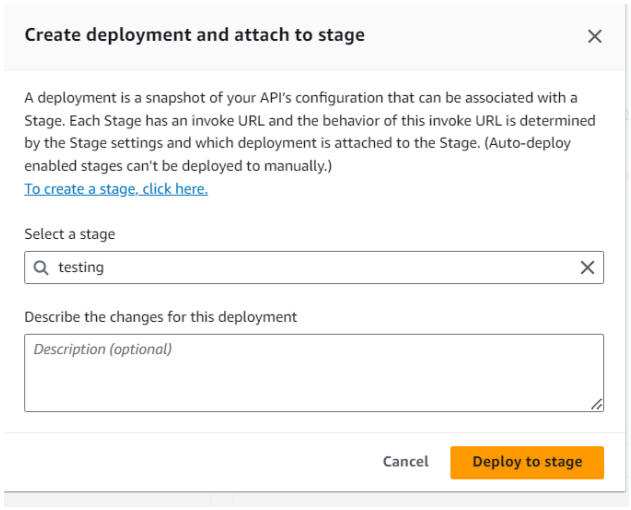


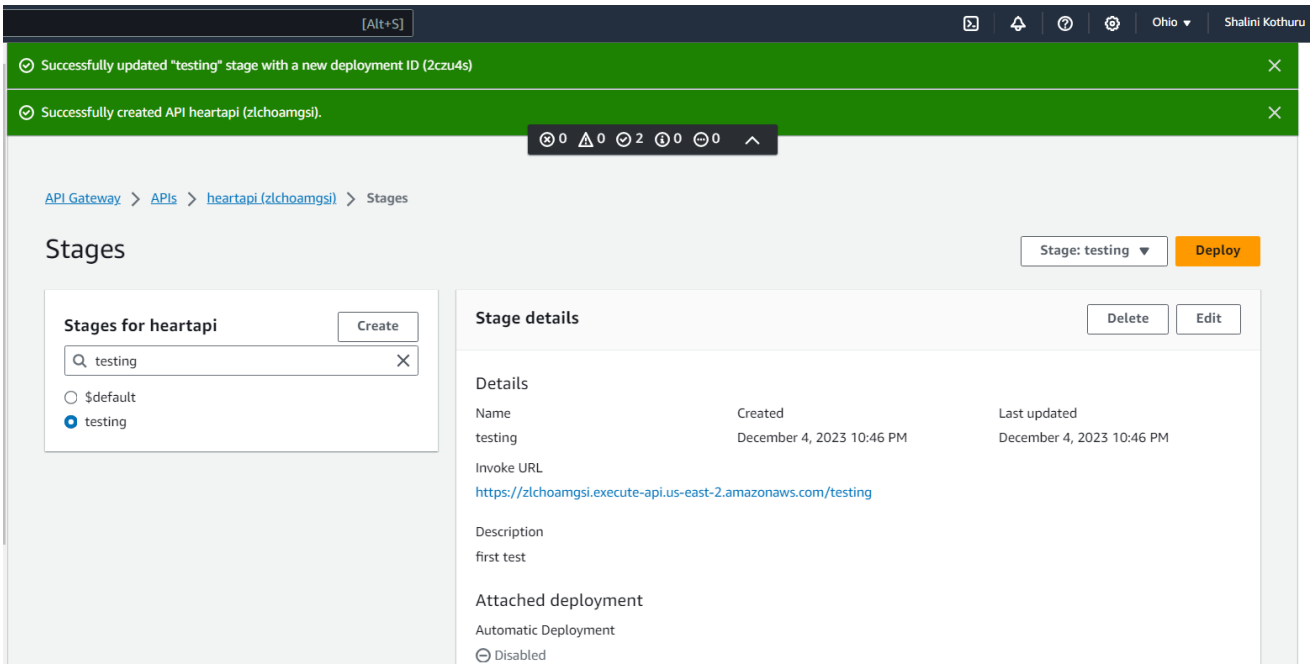
Once I created stage. It gave me one url. This url will be used when developing frontend. Notedown the url and click deploy

URL: https://zlchoamgsi.execute-api.us-east-2.amazonaws.com/testing

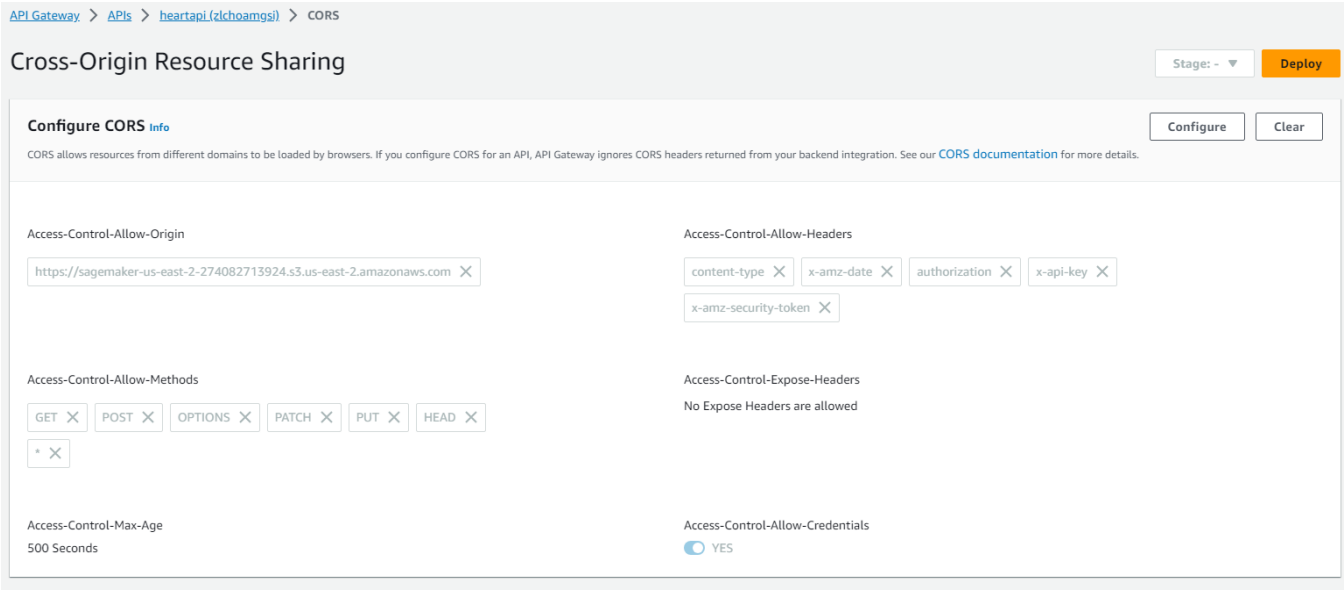


Select the stage you just created and hit deploy to stage.

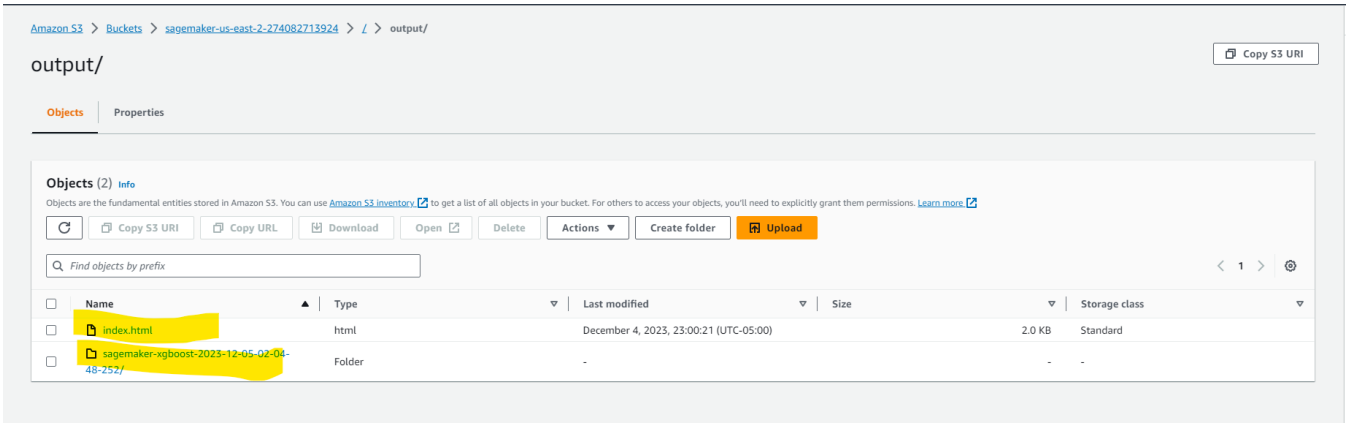




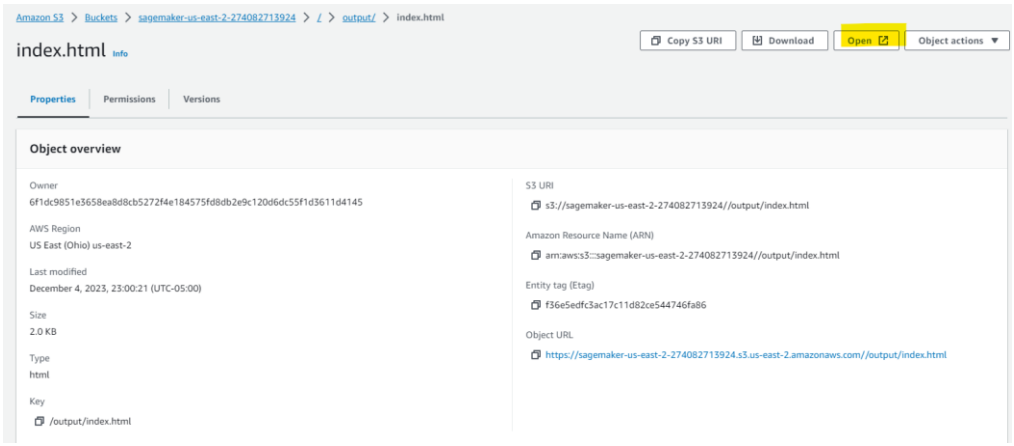
Update access control allow credentials for this API and then deploy it again



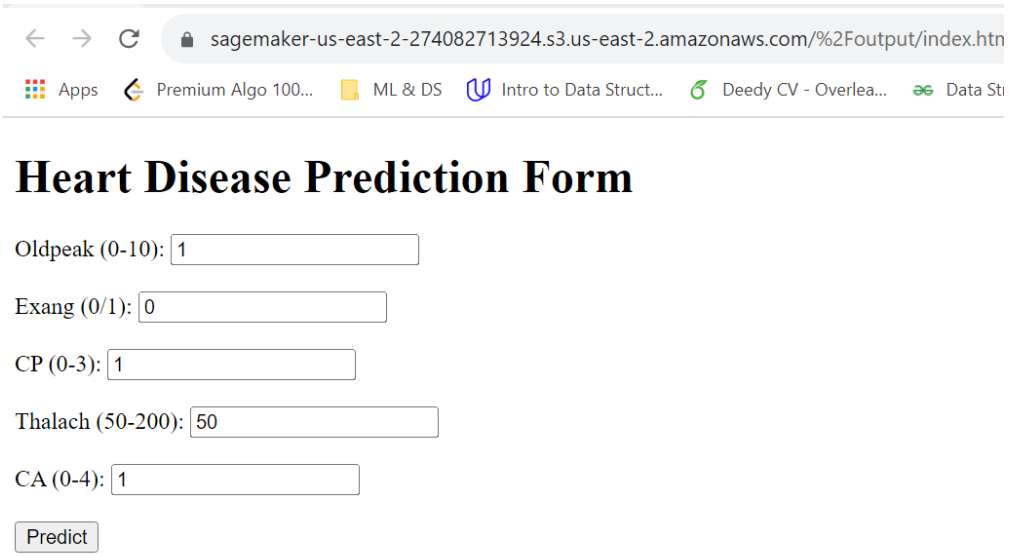
Now go to S3 to store the html file. I have store html file at the same location where I stored sagemaker endpoint



Now click on index.html. It will give details of index.html. Now click on static website URL to access your web application. You can do it by clicking on open button highlighted in below screenshot



It will redirect to following web page. As the goal of this assignment is not about the developing the web application with good UI, I did basic web page. Enter details and click on predict



On clicking predict, it gives the presence of heart disease basically the target variable(0 = no disease or 1 = disease) as show in screenshot.