Coding Assessment

1. Develop a simple website:

- **Frontend: Use React.js to build a simple web page (whatever you like to present). Just like to see if you know how to use components with React.**
- **Backend: Use Django to build at least two API endpoints for the frontend page you created to use.**

2. Upload the assignment to a GitHub repo:

- **Please submit a pdf file that contains the link of the repo and description of your website.**

3. (Bonus) (Deployment) Draw a 3-tier AWS architecutre diagram based on the website you write:

- **Reference: https://aws.amazon.com/architecture/icons/**
- **Please include this diagram as well as the description to the pdf file if you plan to complete this bonus question.**

Solutions:

2. GitHub repo:
https://github.com/ArunRao1997/TasKManagerApp

Access the Deployed React App:
https://arunrao1997.github.io/TasKManagerApp/

Also View My Past Projects:
GitHub repo:https://github.com/ArunRao1997/imdb

Access the Deployed React App: IMDB
https://arunrao1997.github.io/imdb/

Description:

Task Manager Application

The Task Manager App is a simple user-friendly tool designed to help users efficiently manage and track their daily tasks. Developed using Python Django for the backend and React for the frontend, this application provides a seamless experience for creating, updating, deleting and viewing tasks.

Features:

1. Create Tasks
2. Update Tasks
3. Delete Tasks
4. View Tasks

EndPoints:

GetAllTasks:

Method: GET

Url: http://localhost:8000/api/tasks/

Response Body:


```
[
  {
    "id": 1,
    "taskName": "Example Task Update1",
    "taskDescription": "This is a task description Changed1."
  },
  {
    "id": 4,
```

"taskName": "Example Task12",

"taskDescription": "This is a task description1."

},

{

"id": 6,

"taskName": "Arun23",

"taskDescription": "Task23"

},

{

"id": 7,

"taskName": "BuyTwoChocolates",

"taskDescription": "You are given an integer array prices representing the prices of various chocolates in a store. You are also given a single integer money, which represents your initial amount of money.\n\nYou must buy exactly two chocolates in such a way that you still have some non-negative leftover money. You would like to minimize the sum of the prices of the two chocolates you buy.\n\nReturn the amount of money you will have leftover after buying the two chocolates. If there is no way for you to buy two chocolates without ending up in debt, return money. Note that the leftover must be non-negative."

},

{

"id": 9,

"taskName": "TestArun3",

"taskDescription": "Desc3456"

},

{

"id": 10,

"taskName": "Jaswant23",

"taskDescription": "TestJ"

```
  },
  {
    "id": 11,
    "taskName": "BreakTheBank",
    "taskDescription": "TestIfBreakable"
  }
]
```


GetTaskById:

Method: GET

Url: http://localhost:8000/api/tasks/7

Response Body:


```
{
  "id": 7,
  "taskName": "BuyTwoChocolates",
  "taskDescription": "You are given an integer array prices representing the prices of various chocolates in a store. You are also given a single integer money, which represents your initial amount of money.\n\nYou must buy exactly two chocolates in such a way that you still have some non-negative leftover money. You would like to minimize the sum of the prices of the two chocolates you buy.\n\nReturn the amount of money you will have leftover after buying the two chocolates. If there is no way for you to buy two chocolates without ending up in debt, return money. Note that the leftover must be non-negative."
}
```

CreateTask:

Method: POST

Url:http://localhost:8000/api/tasks/

Payload: {

"taskName": "Example Task1",

"taskDescription": "This is a task description1."

}

Response Body: {

"taskName": "Example Task1",

"taskDescription": "This is a task description1."

}

UpdateTask:

Method: PUT

Url:http://localhost:8000/api/tasks/1/

Payload:{

"taskName": "Example Task Update1",

"taskDescription": "This is a task description Changed1."

}

Response Body: {

"taskName": "Example Task Update1",

"taskDescription": "This is a task description Changed1."

}

DeleteTaskById:

Method: DELETE

Url:http://localhost:8000/api/tasks/1/

Response Body: {

"taskName": "Example Task Update1",

"taskDescription": "This is a task description Changed1."

}

Tech Stack:

Backend: Python Django

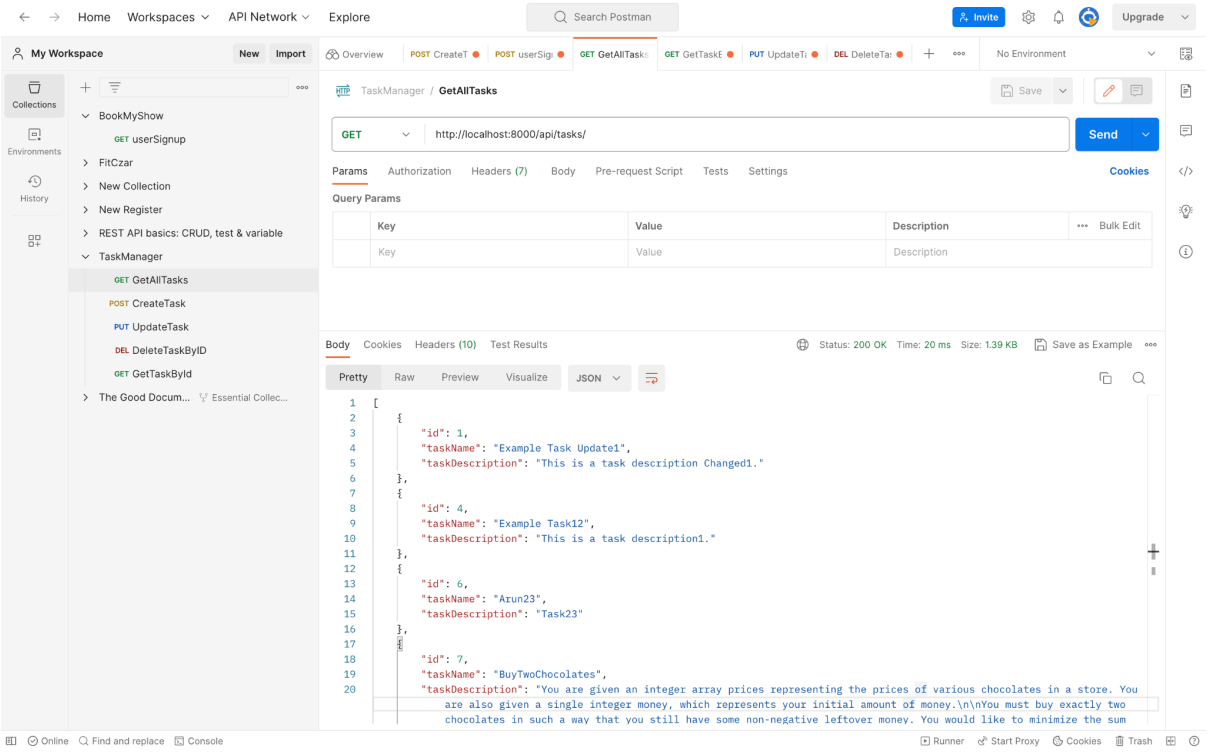
Frontend: React

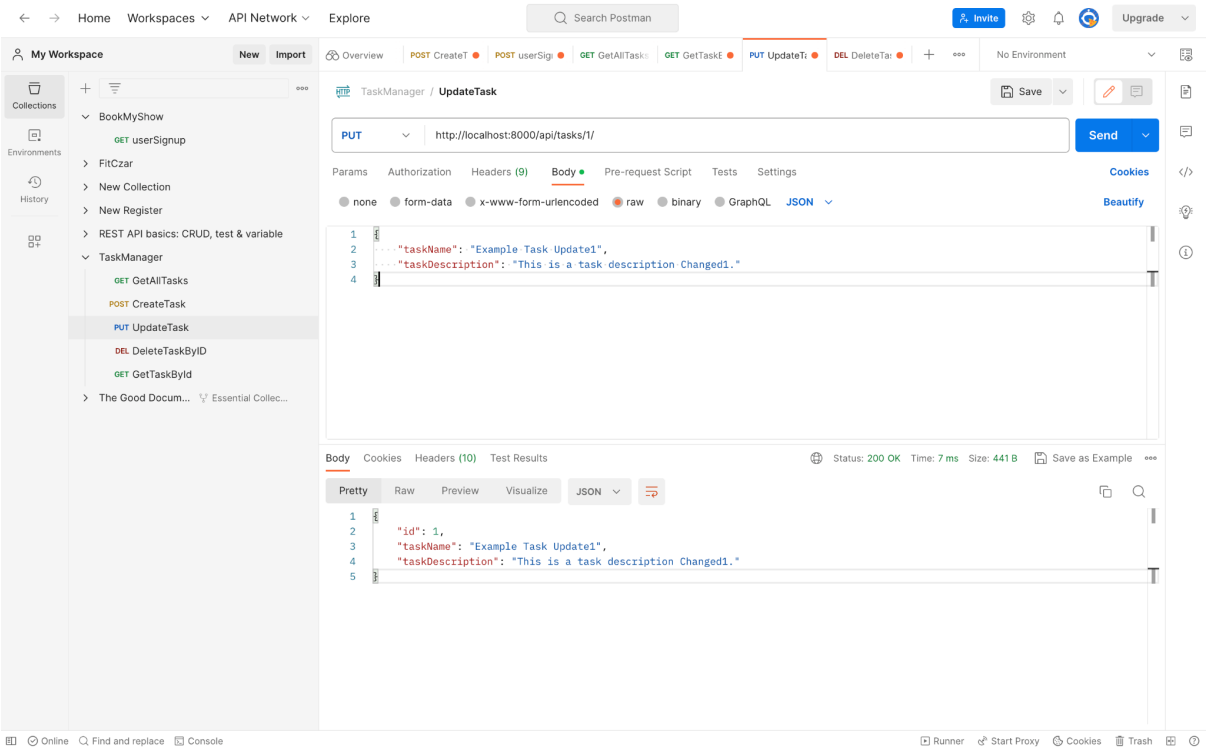
Database: SQLite3

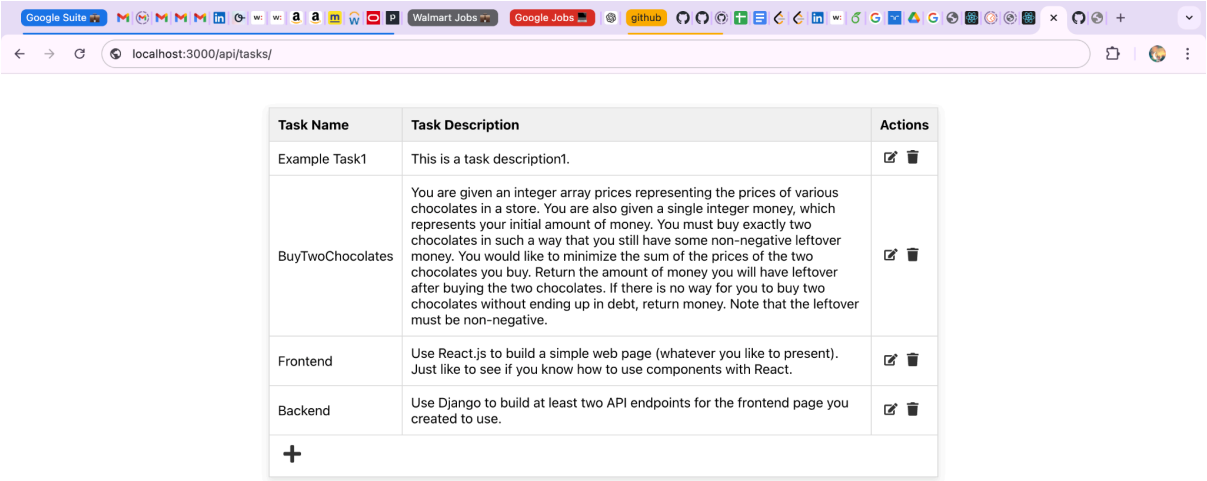
Version Control: Git

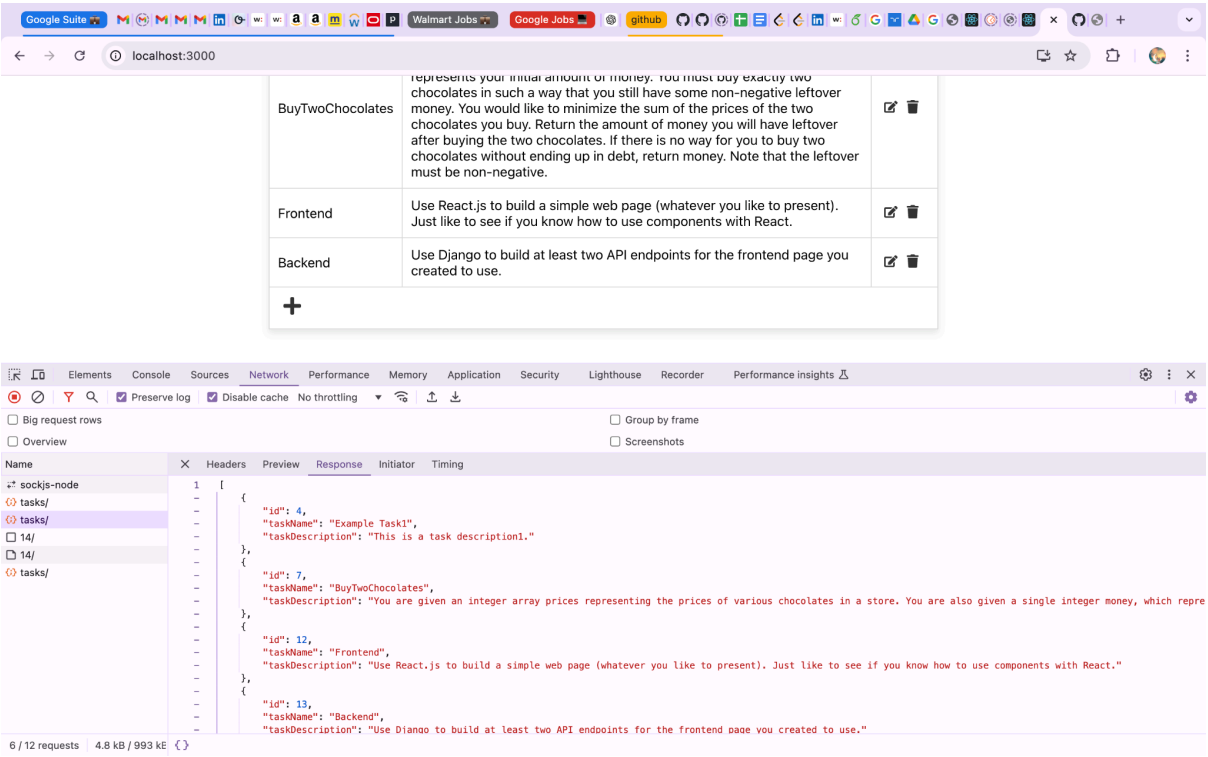
Deployment: GitHub Pages (Frontend)

Snapshots demonstrating the work done:

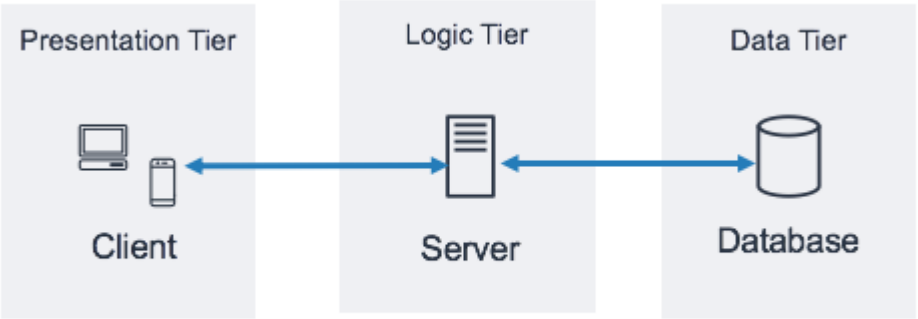


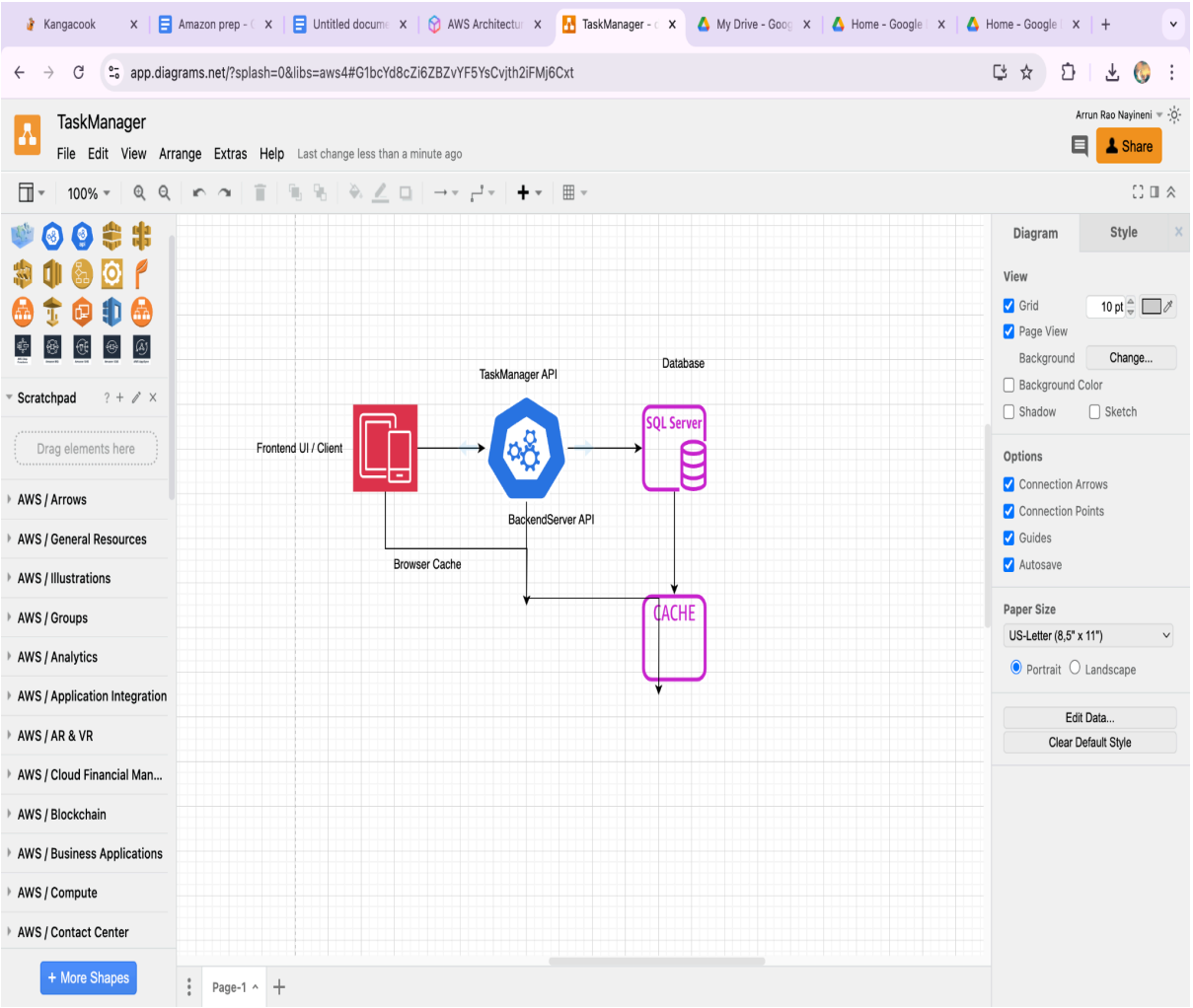






3. 3-tier AWS architecutre diagram


Presentation Tier
Client
Logic Tier
Server
Data Tier
Database


app.diagrams.net/?splash=0&libs=aws4#G1bcYd8cZi6ZBZvYF5YsCvjth2iFMj6Cxt
TaskManager
File Edit View Arrange Extras Help
Last change less than a minute ago
Share
Frontend UI / Client
TaskManager API
Database
SQL Server
BackendServer API
Browser Cache
CACHE
Scratchpad
Drag elements here
AWS / Arrows
AWS / General Resources
AWS / Illustrations
AWS / Groups
AWS / Analytics
AWS / Application Integration
AWS / AR & VR
AWS / Cloud Financial Man...
AWS / Blockchain
AWS / Business Applications
AWS / Compute
AWS / Contact Center
+ More Shapes
Page-1
Diagram
Style
View
Grid
Page View
Background
Change...
Background Color
Shadow
Sketch
Options
Connection Arrows
Connection Points
Guides
Autosave
Paper Size
US-Letter (8,5" x 11")
Portrait
Landscape
Edit Data...
Clear Default Style