MIDDLE EAST TECHNICAL UNIVERSITY
NORTHERN CYPRUS CAMPUS

Computer Engineering Program

# CNG 495
# CLOUD COMPUTING

## Fall 2024 – Term Project Second Step Progress Report

## ClearTasks

**Team Member:**

**Raed H. Manna – 2550911**

**Omar A. Mourad - 2487080**

# Contents


..... 1

**1. Introduction:** ..... 4

**2. Milestones Achieved** ..... 4

**2.1 Individual Contributions:** ..... 4

**2.2 Timeline of Achievements:** ..... 5

**2.3 Screenshots and Figures:** ..... 5

**3. Cloud Technologies and Tutorials** ..... 10

**3.1 Cloud Model (PaaS):** ..... 10

**3.2 AWS Services Setup:** ..... 10

**4. Milestones Remaining** ..... 11

**4.1 Planned Tasks and Responsibilities** ..... 11

**5. GitHub Repository Structure** ..... 11

**6. Difficulties Faced** ..... 12

**7. References** ..... 12

## Table of Figures:

**Figure 1: commit history** .....5
**Figure 2: cloud configuration keys** .....6
**Figure 3: Login/Registration FrontEnd** .....7
**Figure 4: Backend Enpoints for login & registration** .....8
**Figure 5:Early Backend Enpoints for posting tasks** .....9

# 1. Introduction:

The **ClearTasks** project is a cloud-based task management solution designed to provide seamless task tracking, secure authentication, and scalable backend services. This report highlights the work we completed so far, the challenges faced, and the plan for completing the remaining tasks. The project is implemented using **AWS (Amazon Web Services)** as the cloud platform, adopting a **Platform as a Service (PaaS)** model.

# 2. Milestones Achieved

## 2.1 Individual Contributions:

- **Raed Manna**
  - **November 22-23:** Set up the initial Git repository and finalized the project structure.
  - **November 25:** Developed the backend setup using **FastAPI**, using AWS Cognito for authentication, S3 for storage, and SNS for notifications.
  - **November 29- December 1st:** Created the React frontend, implemented the login and registration components, and connected them to the backend API & AWS.

- **Omar A. Mourad**
  - **November 28:** Implemented config.py to integrate cloud services with backend logic, connecting FastAPI to AWS Cognito, S3, and SNS.
  - **November 29:** Worked on backend task APIs and database setup.
  - **November 30:** Implemented the main dashboard for the web app and integrated it with the API

- **Both:**
  - **November 24:** Created an AWS account and configured three cloud services: **Cognito**, **S3**, and **SNS**. Generated access keys to implement configurations. Set up the .env file to manage these credentials securely.

## 2.2 Timeline of Achievements:

| Date/Week | Milestone | Responsible Member |
|---|---|---|
| November 22-23 | Initial Git repository structure setup | Raed |
| November 24 | AWS account setup and **.env** configuration | Omar & Raed |
| November 25 | Backend setup using **FastAPI** | Raed |
| November 28 | Backend **config.py** implementation | Omar |
| November 29 | **React** frontend with login component | Raed |
| November 30 | Main **Dashboard** implementation | Omar |
| December 1st | Registration Endpoint+ Frontend refined | Raed |

## 2.3 Screenshots and Figures:

- **Figure 1:** GitHub commit history illustrating milestones we mentioned above.

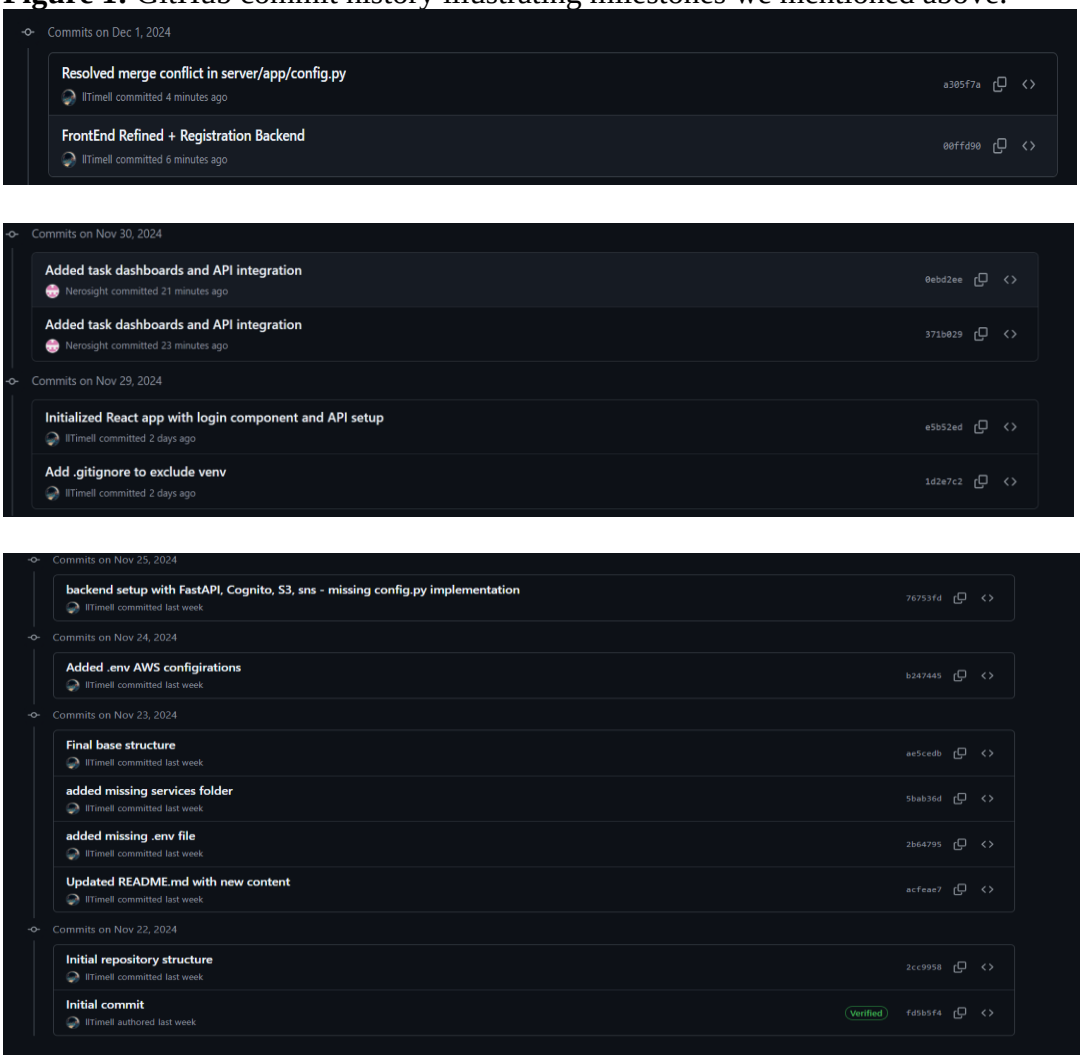


***Figure 1: commit history***

**Note: IITimeII → Raed H. Manna, Nerosight→ Omar A. Mourad**

- **Figure 2:** Content of .env file (AWS credentials, securely managed).

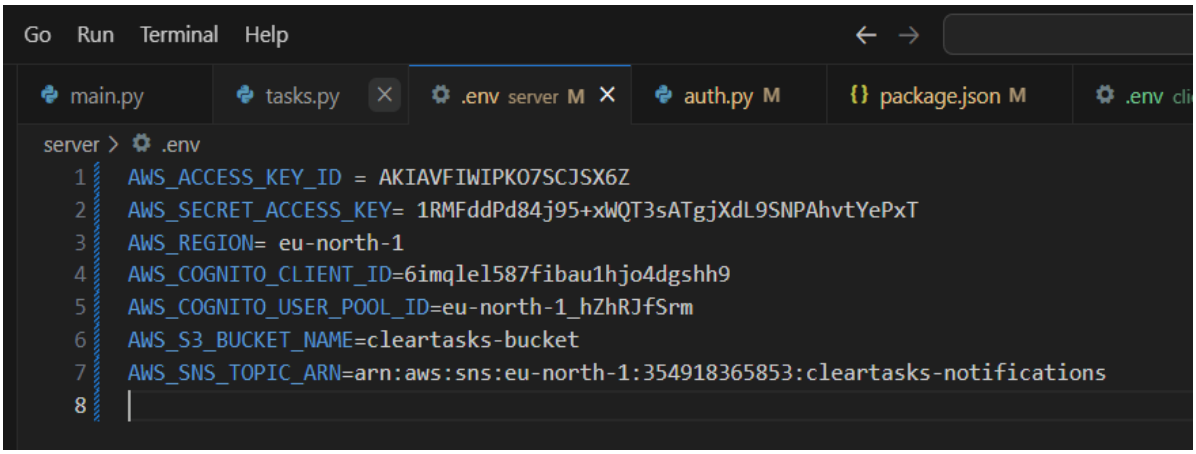


**Figure 2: cloud configuration keys**

- **Figure 3:** Initial React UI screenshot with a login/registration form.

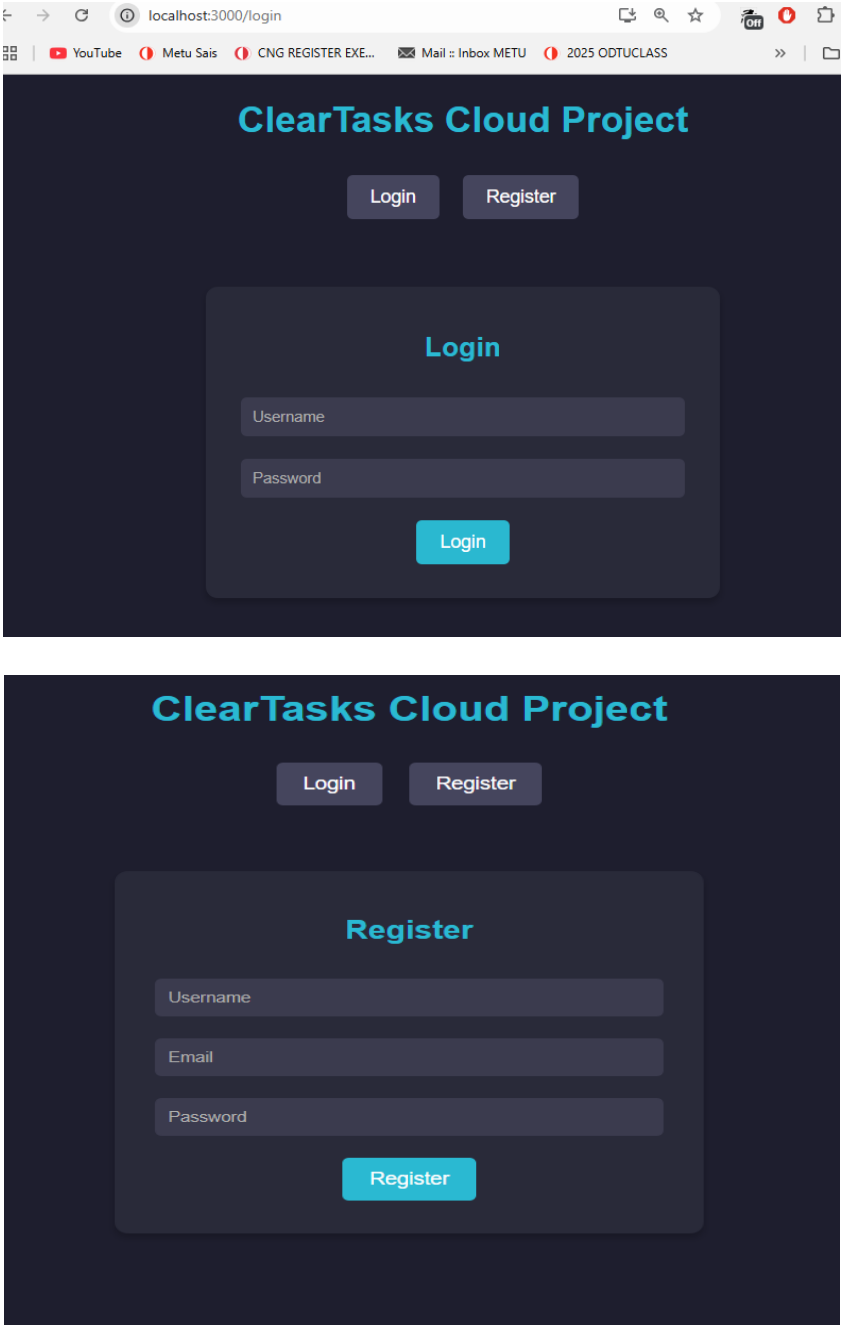


***Figure 3: Login/Registration FrontEnd***

- **Figure 4:** Backend Endpoints for login and registration- connected to **cognito**.

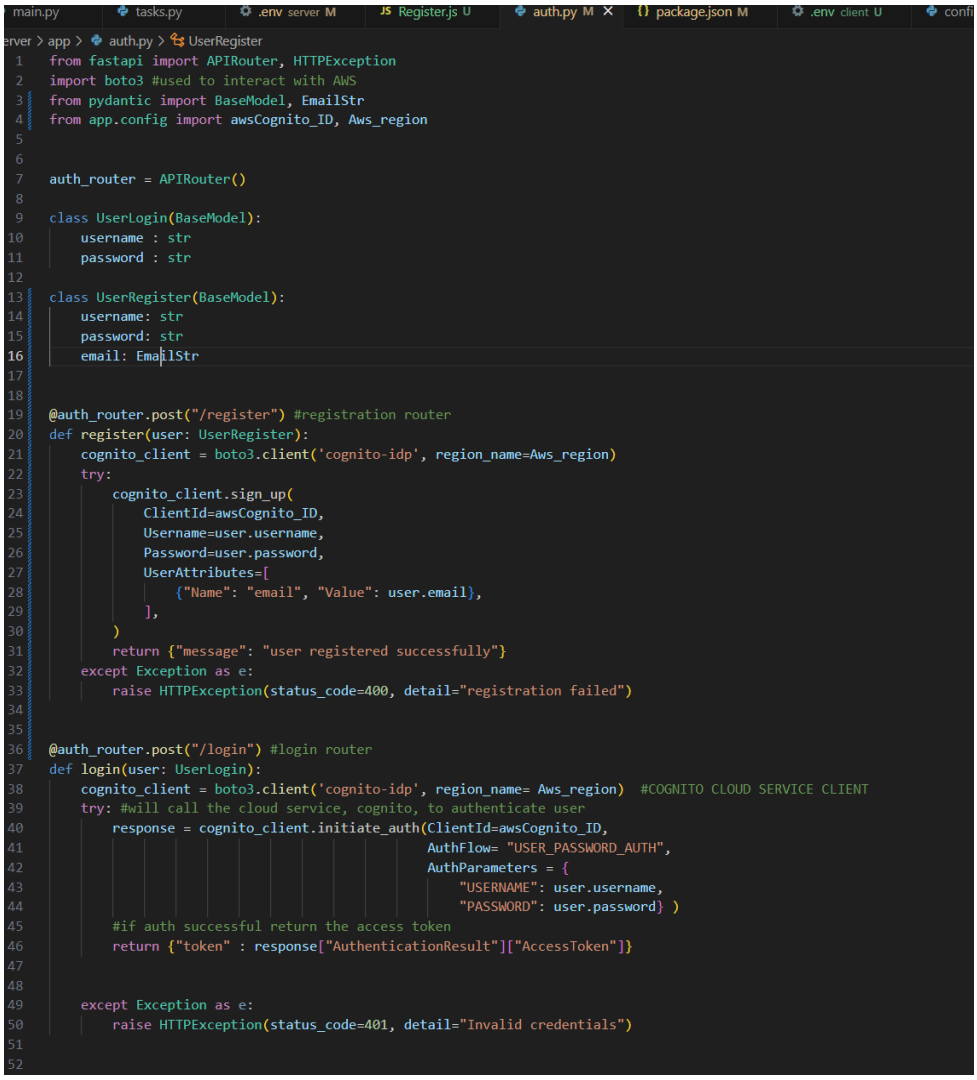


***Figure 4: Backend Enpoints for login & registration***

- **Figure 5:** Early Backend Endpoints for main feature (tasks)- connected to **s3** and **sns**.

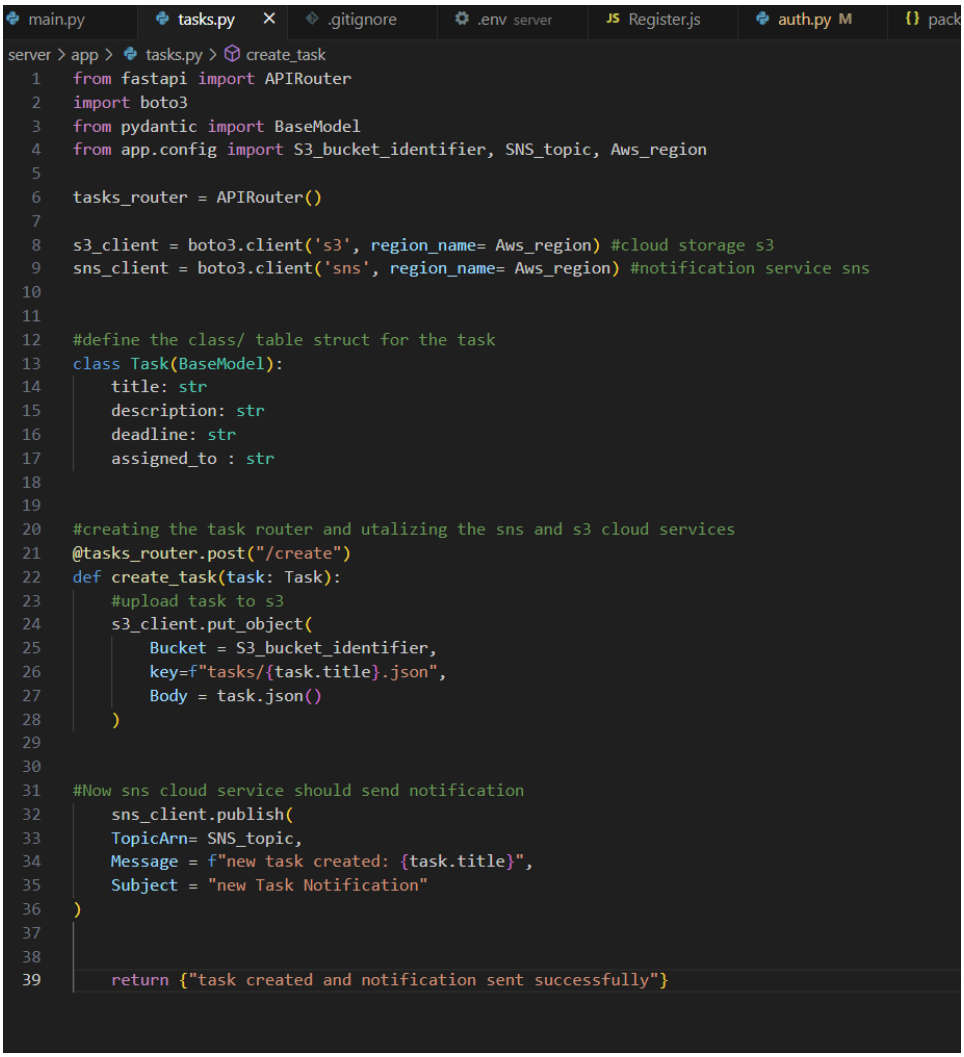


***Figure 5:Early Backend Enpoints for posting tasks***

# 3. Cloud Technologies and Tutorials

## 3.1 Cloud Model (PaaS):

The project uses a **Platform as a Service (PaaS)** model. We chose this model because it allows us to focus on application development without worrying about underlying infrastructure. PaaS enables easy deployment of applications and simplifies the management of cloud resources.

- **AWS Cognito:** Handles user authentication, user pools, and secure access to APIs.
- **AWS S3:** Used for storing static assets (e.g., files, tasks) uploaded by users.
- **AWS SNS:** Used for sending notifications to users when tasks are created, updated, or completed.

## 3.2 AWS Services Setup:

1. **AWS Cognito:**
    - Created a **User Pool** to manage user sign-ups and sign-ins.
    - Configured multi-factor authentication (MFA) for enhanced security.
2. **AWS S3:**
    - Created a bucket named cleartasks-bucket.
    - Configured CORS policies to allow cross-origin requests from the frontend.
3. **AWS SNS:**
    - Created a topic task-updates for task-related notifications.
    - Subscribed users to the topic for real-time updates.

# 4. Milestones Remaining

## 4.1 Planned Tasks and Responsibilities

| Date/Week | Milestone |
|---|---|
| December 2 – December 8 | Add missing backend Endpoints for the main features |
| December 9 - December 15 | Complete the main features in task.py + refine FrontEnd |
| December 16 - December 22 | Final integration, testing and bug fixes |

# 5. GitHub Repository Structure

The general repository structure is as follows: **"Details in the repository->README.md"**

/client "FrontEnd – React"
  /src
    /components
      - Login.js
      -Register.js
  - package.json

/server "Backend – FastAPI"
  /app
    -auth.py
    -tasks.py
    -config.py
    -main.py
   -.env
   - requirements.txt

/docs
  - Proposal.pdf
  - Progress_Report.pdf

README.md

- **README.md:** explains structure in more details. (Please expand it, so it shows in an organized way).
- **.env:** contains environment variables for cloud services.

**GitHub URL:** https://github.com/llTimell/ClearTasks.git

## 6. Difficulties Faced

1. **IDE Compatibility:**
   Initially, we used **PyCharm** for development. However, upon starting the React frontend implementation, we discovered that PyCharm lacked the necessary plugins for React and JavaScript development. This made us switch to **VS Code**, which caused some workflow disruptions; especially when it came to committing after cloning the repository.

2. **GitHub Repository Management:**
   After the first commit after the change mentioned above, several issues arose due to a lack of familiarity with advanced Git commands. This required additional research on reverting commits, dealing with untracked files, and handling. gitignore rules to prevent unintended file tracking (e.g., venv).

3. **Cloud Service Configuration:**
   Setting up AWS services such as Cognito and S3 required us to read more about what each service has to offer and what does it actually do, or mean. For example, what is a User Pool?, what is a bucket?

## 7. References


1. AWS Documentation: Cognito, S3, SNS
2. ReactJS Documentation: https://reactjs.org/
3. FastAPI Documentation: https://fastapi.tiangolo.com/
4. PyCharm IDE: https://www.jetbrains.com/pycharm/
5. Visual Studio Code (VS Code): https://code.visualstudio.com/