



**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:

<b><u>Figure 1: commit history</u></b> .....	5
<b><u>Figure 2: cloud configuration keys</u></b> .....	6
<b><u>Figure 3: Login/Registration FrontEnd</u></b> .....	7
<b><u>Figure 4: Backend Endpoints for login &amp; registration</u></b> .....	8
<b><u>Figure 5:Early Backend Endpoints for posting tasks</u></b> .....	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:

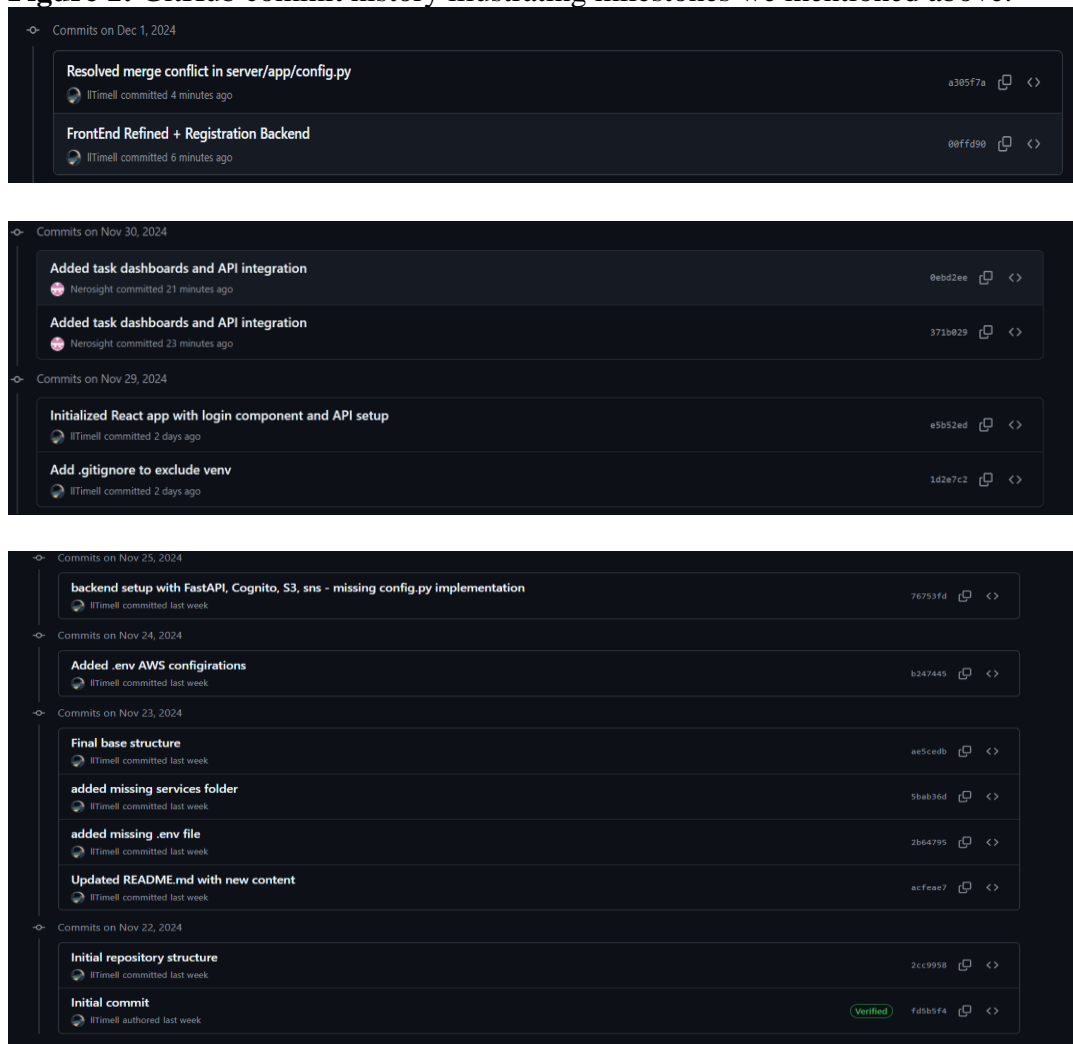
- **Raed Manna**
  - **November 22-23:** Set up the initial Git repository and finalized the project structure.
  - **November 25:** Developed the backend setup using **FastAPI**, using AWS Cognito for authentication, S3 for storage, and SNS for notifications.
  - **November 29- December 1st:** Created 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 <b>.env</b> configuration	Omar & Raed
November 25	Backend setup using <b>FastAPI</b>	Raed
November 28	Backend <b>config.py</b> implementation	Omar
November 29	<b>React</b> frontend with login component	Raed
November 30	Main <b>Dashboard</b> implementation	Omar
December 1st	Registration Endpoint+ Frontend refined	Raed

## 2.3 Screenshots and Figures:

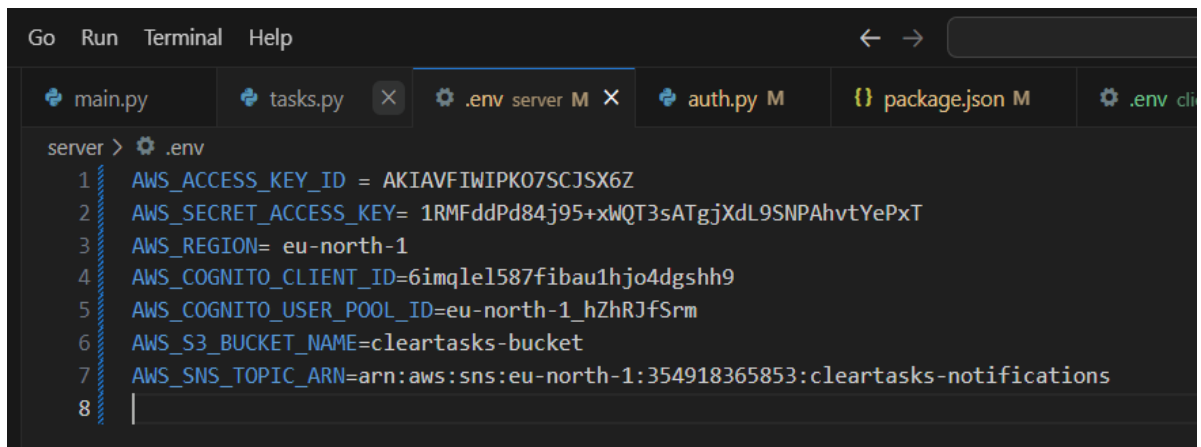
- **Figure 1:** GitHub commit history illustrating milestones we mentioned above.



**Figure 1: commit history**

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

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



The image shows a code editor window with a dark theme. The top bar includes 'Go', 'Run', 'Terminal', and 'Help' menus. Below the menu bar, there are several tabs: 'main.py', 'tasks.py', '.env server M', 'auth.py M', 'package.json M', and '.env cli'. The '.env server M' tab is active, showing the following content:

```
server > .env
1 AWS_ACCESS_KEY_ID = AKIAVFIWIPK07SCJSX6Z
2 AWS_SECRET_ACCESS_KEY= 1RMFddPd84j95+xwQT3sATgjXdL9SNPAhvtYePxT
3 AWS_REGION= eu-north-1
4 AWS_COGNITO_CLIENT_ID=6imqlel587fibau1hjo4dgshh9
5 AWS_COGNITO_USER_POOL_ID=eu-north-1_hZhRJfSrm
6 AWS_S3_BUCKET_NAME=cleartasks-bucket
7 AWS_SNS_TOPIC_ARN=arn:aws:sns:eu-north-1:354918365853:cleartasks-notifications
8
```

**Figure 2: cloud configuration keys**

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

The figure consists of two screenshots of a web application interface. Both screenshots show a dark blue background with the title "ClearTasks Cloud Project" in a light blue font at the top. Below the title are two buttons: "Login" and "Register".

The top screenshot shows the "Login" form. It has a title "Login" in light blue. Below the title are two input fields: "Username" and "Password". At the bottom of the form is a "Login" button in light blue.

The bottom screenshot shows the "Register" form. It has a title "Register" in light blue. Below the title are three input fields: "Username", "Email", and "Password". At the bottom of the form is a "Register" button in light blue.

**Figure 3: Login/Registration FrontEnd**

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

```

main.py tasks.py .env server M JS Register.js U auth.py M X () package.json M .env client U conli
server > app > auth.py > UserRegister
1 from fastapi import APIRouter, HTTPException
2 import boto3 #used to interact with AWS
3 from pydantic import BaseModel, EmailStr
4 from app.config import awsCognito_ID, Aws_region
5
6
7 auth_router = APIRouter()
8
9 class UserLogin(BaseModel):
10     username : str
11     password : str
12
13 class UserRegister(BaseModel):
14     username: str
15     password: str
16     email: EmailStr
17
18
19 @auth_router.post("/register") #registration router
20 def register(user: UserRegister):
21     cognito_client = boto3.client('cognito-idp', region_name=Aws_region)
22     try:
23         cognito_client.sign_up(
24             ClientId=awsCognito_ID,
25             Username=user.username,
26             Password=user.password,
27             UserAttributes=[
28                 {"Name": "email", "Value": user.email},
29             ],
30         )
31         return {"message": "user registered successfully"}
32     except Exception as e:
33         raise HTTPException(status_code=400, detail="registration failed")
34
35
36 @auth_router.post("/login") #login router
37 def login(user: UserLogin):
38     cognito_client = boto3.client('cognito-idp', region_name= Aws_region) #COGNITO CLOUD SERVICE CLIENT
39     try: #will call the cloud service, cognito, to authenticate user
40         response = cognito_client.initiate_auth(ClientId=awsCognito_ID,
41             AuthFlow= "USER_PASSWORD_AUTH",
42             AuthParameters = {
43                 "USERNAME": user.username,
44                 "PASSWORD": user.password } )
45         #if auth successful return the access token
46         return {"token" : response["AuthenticationResult"]["AccessToken"]}
47
48
49     except Exception as e:
50         raise HTTPException(status_code=401, detail="Invalid credentials")
51
52

```

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



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

```

main.py tasks.py .gitignore .env server JS Register.js auth.py M pack
server > app > tasks.py > create_task
1  from fastapi import APIRouter
2  import boto3
3  from pydantic import BaseModel
4  from app.config import S3_bucket_identifier, SNS_topic, Aws_region
5
6  tasks_router = APIRouter()
7
8  s3_client = boto3.client('s3', region_name= Aws_region) #cloud storage s3
9  sns_client = boto3.client('sns', region_name= Aws_region) #notification service sns
10
11
12 #define the class/ table struct for the task
13 class Task(BaseModel):
14     title: str
15     description: str
16     deadline: str
17     assigned_to : str
18
19
20 #creating the task router and utalizing the sns and s3 cloud services
21 @tasks_router.post("/create")
22 def create_task(task: Task):
23     #upload task to s3
24     s3_client.put_object(
25         Bucket = S3_bucket_identifier,
26         key=f"tasks/{task.title}.json",
27         Body = task.json()
28     )
29
30
31 #Now sns cloud service should send notification
32     sns_client.publish(
33         TopicArn= SNS_topic,
34         Message = f"new task created: {task.title}",
35         Subject = "new Task Notification"
36     )
37
38
39     return {"task created and notification sent successfully"}

```

**Figure 5:**Early Backend Endpoints 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 clear-tasks-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/>