Task-1

Suppose you have this email address “Amit_ml@gmail.edu”

- Input Validation: Check if the input string contains exactly one "@" symbol and at least one "." after the "@" symbol. If it's not a valid email, return "Invalid email".

- Extract Username: Extract and return the part of the email before the "@" symbol.

- Extract Domain: Extract and return the domain (the part between "@" and the last ".").

- Check for Domain Ending: Check if the email ends with ".com". If it does, return "Commercial Domain". If it ends with ".edu", return "Educational Domain". Otherwise, return "Other Domain".

Task-2:

Encoded Message:
`###!!@mocleW EPGTQ!!!6789`

Steps to Decode:

1. Extract the core part of the message: `"mocleW EPGTQ"`.
2. Reverse the first word: `"mocleW"` becomes `"Welcome"`.
3. Replace shifted vowels in the second word:
    - `"EPGTQ"`: No vowels to change.
4. Final decoded message: `"Welcome PGTQ"`.

Task-3:

Encoded Message:
`&&&**$gnirtS PLIO!!@1234`

Steps to Decode:

1. Extract the core part of the message: `"gnirtS PLIO"`.
2. Reverse the first word: `"gnirtS"` becomes `"String"`.
3. Replace shifted vowels in the second word:
    - `"PLIO"`: Replace `I->E` and `O->U` to get `"PLEU"`.
4. Final decoded message: `"String PLEU"`.

Task-4:

Encoded Message:
`##$$$@!yalpstcejorp EPUVT****9887`

Steps to Decode:

1. Extract the core part of the message: `"yalpstcejorp EPUVT"`.
2. Reverse the first word: `"yalpstcejorp"` becomes `"projectplay"`.
3. Replace shifted vowels in the second word:
    - `"EPUVT"`: Replace `E->A`, `U->O` to get `"APOVT"`.
4. Final decoded message: `"projectplay APTOV"`.

Task-5: Create a New GitHub Repository and Set Up Your Project Structure

Objective:

You will create a new GitHub repository named `Amit` and set up the following folder structure for a Python project focused on Machine Learning.

Instructions:

1. **Create a New GitHub Repository**:
   - Go to GitHub.
   - Log in to your account.
   - Click on the "+" icon in the top right corner and select "New repository."
   - Name your repository `Amit`.
   - Add a description (optional) and choose whether to make it public or private.
   - Click on "Create repository."
2. **Clone the Repository to Your Local Machine**:
   - Open VSCode.
   - Open the terminal in VSCode (View > Terminal).
   - Run the following command to clone your repository (replace `<your-username>` with your GitHub username):

     ```
     bash
     Copy code
     git clone https://github.com/<your-username>/Amit.git
     ```

   - Navigate into the cloned repository:

     ```
     bash
     Copy code
     cd Amit
     ```

3. **Create the Folder Structure**:
   - Inside the `Amit` folder, create a folder named `python-for-ml`. You can do this in the terminal:

     ```
     bash
     Copy code
     mkdir python-for-ml
     ```

4. **Create a README File**:
   - In the root of the `Amit` repository, create a file named `README.md`:

     ```
     bash
     Copy code
     touch README.md
     ```

   - Open `README.md` in VSCode and add the following content:

```
markdown
Copy code
# Amit Repository

## Project Overview
This repository is designed for Python projects related to
Machine Learning.

## Folder Structure
- `python-for-ml/`: Contains Python scripts and resources for
Machine Learning.

## Getting Started
To get started with this project, clone the repository and
navigate to the `python-for-ml` folder.
```

5. **Stage and Commit Your Changes**:
   - In the terminal, stage the changes:

```
bash
Copy code
git add .
```

   - Commit the changes with a message:

```
bash
Copy code
git commit -m "Initial setup: Created folder structure and README
file"
```

6. **Push Changes to GitHub**:
   - Push your changes to the GitHub repository:

```
bash
Copy code
git push origin main
```

7. **Verify Your Changes on GitHub**:
   - Go back to your GitHub repository in your browser and refresh the page to see the new folder structure and the README file.

Deliverables:

- A new GitHub repository named `Amit` with the specified folder structure.
- A `README.md` file containing the project overview and instructions.

Bonus:

- Experiment with adding additional Python scripts in the `python-for-ml` folder and document their purposes in the README file.