CS661 Course Project - Temporary README for Team Collaboration

---

🚨 Important Notice

This is a **temporary README** to guide all team members on how to work on this project without breaking anything. We are working on separate modules, and we will combine our work later. **Do NOT make changes directly on the `main` branch.** Follow the steps below carefully to work on your own module.

Project Overview

This repository is for our CS661 Course Project: a Fake News Analysis Dashboard with D3.js visualizations. Each team member works in their own module folder:

- `Module_Piyush_Ravi/`: Our work on the Fake News Analysis Dashboard.
- (Add your module folder here, e.g., `Module_Ananya_Arpita/`)

Later, we will combine all modules into a final project on the `main` branch.

How to Work on This Project (Step-by-Step)

Step 1: Set Up Your Local Environment

1. **Clone the Repository**: Open a terminal on your computer and run:

   ```
   git clone https://github.com/Piyushiitk24/CS661_Course_Project.git
   cd CS661_Course_Project
   ```

   This downloads the project to your computer.

2. **Set Up a Virtual Environment**: We use a virtual environment to manage dependencies. Create and activate one:

   ```
   python3 -m venv venv
   source venv/bin/activate
   ```

   Your terminal should show `(venv)` at the start.

3. **Install Dependencies**: Install the required packages listed in `requirements.txt`:

   ```
   pip install -r requirements.txt
   ```

If you need a new package (e.g., `pandas`), install it and update `requirements.txt`:

```
pip install pandas
pip freeze > requirements.txt
```

Step 2: Create Your Own Module Folder

Each person must work in their own folder to avoid breaking others' work.

1. Create a folder with your name, e.g., `Module_Ananya_Arpita`:

```
mkdir Module_Ananya_Arpita
```

2. Put all your files (code, data, etc.) inside this folder. For example:

```
CS661_Course_Project/
├── Module_Piyush_Ravi/
├── Module_Ananya_Arpita/    ← Your folder
├── data/
├── requirements.txt
├── README.md
└── venv/
```

Step 3: Create Your Own Branch (Do NOT Work on `main`)

To avoid messing up the `main` branch, you must work on your own branch.

1. **Create a New Branch**: Replace `yourname_feature` with your name, e.g., `Ananya_Arpita_feature`:

```
git checkout -b yourname_feature
```

Example for Ananya_Arpita:

```
git checkout -b Ananya_Arpita_feature
```

2. **Verify You're on Your Branch**: Run:

```
git branch
```

You should see your branch name with a * next to it, like * `Ananya_Arpita_feature`.

Step 4: Work on Your Module

- Make changes only in your folder (`Module_Ananya_Arpita/`).
- Add new files, edit code, etc., inside your folder.
- **Do NOT touch `main` or other people's folders** (like `Module_Piyush_Ravi/`).

Step 5: Save and Share Your Work

After making changes, save them to your branch and share them on GitHub.

1. **Stage Your Changes**: Add your files to Git:

   ```
   git add Module_Ananya_Arpita/
   ```

   Or add all changes:

   ```
   git add .
   ```

2. **Commit Your Changes**: Write a clear message about what you did:

   ```
   git commit -m "Add my initial files to Module_Ananya_Arpita"
   ```

3. **Push to GitHub**: Push your branch to GitHub:

   ```
   git push origin Ananya_Arpita_feature
   ```

   Example for Ananya_Arpita:

   ```
   git push origin Ananya_Arpita_feature
   ```

Step 6: Create a Pull Request (PR) to Share Your Work

To share your work with the team (without affecting `main` yet):

1. Go to `https://github.com/Piyushiitk24/CS661_Course_Project`.

2. You'll see a prompt for your branch (e.g., `Ananya_Arpita_feature`). Click "Compare & pull request."
3. Add a description of your changes (e.g., "Added my dashboard code in Module_Ananya_Arpita").
4. Click "Create pull request." **Do NOT merge it yet.** We will combine all work later.

Step 7: Stay in Sync with `main`

To avoid conflicts, pull updates from `main` regularly:

```
git checkout main
git pull origin main
git checkout yourname_feature
git merge main
```

If there are conflicts, ask for help (e.g., message in Whatsapp Group).

Rules to Follow

- **Never work on `main` directly.** Always use your branch.
- **Only work in your own folder.** Don't touch other people's folders (e.g., `Module_Piyush_Ravi/`).
- **Commit often** with clear messages (e.g., "Add login page to Module_Ananya_Arpita").
- **Pull from `main` before starting new work** to stay updated.
- **Ask for help** if you're unsure—don't guess and break things.

How We'll Combine Work Later

- Each person's branch (e.g., `Ananya_Arpita_feature`, `piyush_feature`) will be reviewed via Pull Requests.
- Once everyone's work is ready, we'll merge all branches into `main`.
- We'll resolve any conflicts together as a team.