# Software Development - Final Project - Fall

**Team size:** 2 - 4 students
**Topic:** You must select a topic that interests all members of the group
**Due Date:** January 19, 2024 at 10:00 pm

**Requirements:**

Your team must showcase your proficiency with frontend and backend using all the resources and tools you have learned during the semester:

- Django
- CSS
- JavaScript
- Bootstrap or Foundation (if you are familiar with this tool)
- Postgres
- Relational Databases
- APIs.

The first thing your team must do is to choose a topic, establish roles, and develop a design for your project. The design will change during the development phase.

Then, your team should agree on how your project will be organized and how you will divide the work. Express your plan in a **design document**.

The **design document** is your first deliverable. All your efforts should be directed to doing this job well; do not spend time working on other aspects of the project until design docs have been developed, reviewed, and revised. The teacher will review the design document and give you feedback if something should be implemented differently regarding the API data (e.g., if you are getting too much data or vice versa).

The project manager must create a GitHub repo to save the project and add the team members and the teacher as collaborators.

## Design Document Specifications:

- Project name.
- A list of applications you will implement.
- The components and actions that will be developed in each application.
- Explanation of how each component relates to the others.
- Database design. Diagram showing tables, fields, PKs, FKs, indexes, and relationships between tables.
- Tentative site map for front-end diagram (See info and example here: https://www.outofsightdesigns.com/what-is-a-sitemap/)
    - Represent each page you envision for your site.
    - Show linkages conveying all possible pathways for a user traversing your site.
- A breakdown of the different tasks required to complete this project.
    - Include assignments of each task to each group member.
- Clearly labeled section delineating APIs you will use.
- This document should be a PDF file. You will add it to your GitHub repo in the designated location (see below), and you must hand in a hard copy (see checkpoints deliverables).

## Project Guidelines:

- API data must be accessed using Python - Django, not JavaScript.
- If needed, add a button on your website to add some initial data to your database (get data from API and save it in the database). This is totally fine. If you do it, you must mention it in the README file. So, the user of your website knows how to use it.
- However, you should also have the option when a new record is added to a table while navigating your website; it should use the API to get the data for the newly inserted record at that moment. For example, if your project has a database to store temperatures by city. When you run the website for the first time, you may populate your table cities with some records, get temperature data from the API for those cities, and save it in the database. When the user inserts a new city in the database, your code should get the temperatures for that city from the API and save it in the database at that moment.
- Write your own CSS. You can look at existing CSS to get some inspiration. For this project, it is preferable to have a CSS that is not as elegant as existing ones but has your own code and shows your creativity.
- You must include JavaScript components (or dynamic interfaces). For example, data that changes when a different field is selected. Or any other situation that requires JavaScript.
- You must implement a responsive website.
- **Do not** add the Python virtual environment to GitHub.
- **Do not** add the secrets.json to GitHub. Add a secrets_template.json, so anyone who runs your project knows what data should be included in the secrets file.

## Devlog:

- Devlog allows any group member to see the project's current state at any time.
- The project manager will ensure the devlog is being maintained but will not make all entries.
- The devlog should be a plain text file stored in the specified location (see below).
- When any team member stops working and pushes changes to GitHub, they should update the devlog, explaining what changes have been made. Include errors/bugs discovered (or created).
- Separate devlog entries with a newline.
- The most recent entry should be at the bottom.
- Each entry should begin with the following format: FirstL -- TIMESTAMP\n ( e.g., JessicaN -- 2024-01-04 22:15 )

## Final Deliverables:

- Hardcopy: Final version of your design document.
- Repo Structure:

```
your_global_project_name
        your_project _name
                django_project_name/
                        all_project_files
                        secrets_template.json
                core/
                        all_app_files
                django_app_name_1/
                        all_app_files
                django_app_name_2/
                        all_app_files
                …..
                …..
                manage.py
        requirements_env/
                dev.in
                main.in
        design.pdf
        devlog.txt
        README.md
```

- README.md
    - Clearly visible at the top: **<Project Name>**
    - Roster with roles
    - Description of your website (short summary, NOT your entire design doc)
    - List the APIs you are using (include links)

    - Launch codes (detailed description):
        - How to install: Start by including the commands needed to build and install the packages from your requirements file. Include any additional pre-requirements that should be done to have your website working.
        - How to run: How the user should start using your website, including the Django commands to run your website. For example, a button should be pressed to populate some data in a database.
        - 
- design.pdf
    - Latest/current version of your design document.
    - Revisions since the first version should be noted/explained in devlog.txt

**<u>Subgoals / Checkpoint Deliverables</u>**:

1. **2024-01-09 10:00 pm**
    - Design doc in specified location (electronic)
    - Hardcopy (beginning of class on 01/10)
2. **2024-01-12 08:00 am**
    - Revised design doc (electronic)
    - Hardcopy revised design doc (beginning of class)
    - Devlog updated with summary performed task and any database, sitemap, or design document revisions
3. **2024-01-17 08:00 am**
    - Requirements in place
    - Summary on readme
    - Launch codes in place
    - API mechanism in place
    - Working database
4. **2024-01-19 10:00 pm**
    - The project manager must email to the teacher with the subject **Final Project - Your Project Title**. Include the repository GitHublink. Attach these files: design.pdf, devlog.txt, README.md. CC'ed all the teammates.

**<u>Reminder:</u>**

If you have questions during the development of this project, please ask the teacher. Do not wait until the last minute to ask questions.

If you are having issues, try to research about it. Then, try to **POST IT ON PIAZZA**. Your peers or instructor will help.

**GOOD LUCK!!!**