Software Development - Final Project - Spring

Team size: 4 students
Topic: You must select a topic that interests all members of the group
Points: 100
Due Date Final Product: June 12, 2024 at 10:00 pm

Requirements:

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

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

The first thing your team must do is to choose a topic. You must choose an interesting topic, such as weather, mta, real estate, ticketmaster, twitter… (this is just for inspiration, you are going to implement a simpler website than those, you only have 3 weeks). You must double check that there is data available through an API for the topic you choose.

Important: Your topic must be approved. Please share the topic with the teacher as soon as you have it.

Then, your team should agree on how your project will be organized and how you will divide the work. Do not split tasks by backend, frontend, or database. Everyone must participate in implementing everything (backend, frontend). You may split the tasks by functionality/feature/application. Express your plan in the **design document**.

The **design document** is your first deliverable. All your efforts should be directed to doing this job well.

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

The group must use branches, one per each teammate. It is **REQUIRED**. If you do not use branches your project will be penalized.

Design Document Specifications:

- Names.
- Group Name (find a cool name for your group).
- 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:
    - 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 neat and well-organized PDF file. You must include the database diagram that could be automatically generated using Graphviz (Python package). It will get the info from your models to generate the graph. You may also use an online tool to generate the database diagram as well, such as drawsql. However, GraphViz will be easier and save you time because it does the job for you.
  You will add this document to your GitHub repo in the designated location (see below), and you must hand in a hard copy (see checkpoints deliverables). I will grade the hard copy, so **DO NOT FORGET TO PRINT AND STAPLE YOUR DOCUMENT.**

Project Guidelines:

- You must design a database to store the data that will be used by your website.
- Implement user authentication. The user needs to be logged in to use your website.
- You may write your own CSS or use an existing CSS (include the source where you are getting the CSS in the design document).
- The frontend should be implemented using Vue.js. It is not necessary for all pages, but at least one, probably the one with more user interaction. It must be well implemented.
- You must implement a responsive website.
- API data must be accessed using Python—Django, not JavaScript. You can do this using a script, directly in a view on the backend or by adding it to your website to insert the API data into your database. Anything you do to populate the database must be included in the README file so that your website's users know how to use it.
- Your website must have a section where you analyze the data you have from the API, which is saved in the database. You must get the data from the database and store it in dataframes. Then, you are going to create charts with the data. Display some data and charts on your webpage. You must add some text on the page explaining the data and charts you are showing. This is your data analysis.

- **Do not** add the Python and Node.js virtual environments 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
                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
        your_project _name_vue
                src
                        apps
                                all_you_vue_apps
                        ..…
                ..…
                ..…
```

```
requirements_env/
        dev.in
        main.in
design.pdf
devlog.txt
README.md
```

- README.md
    - Clearly visible at the top: **<Project Name>**
    - Names
    - Group Name
    - 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:** Include the commands needed to build and install the packages from your requirements file. Also, include any additional pre-requirements that should be done to get your website working.
        - **How to run:** This section describes how the user should start using your website, including the Django commands to run it. For example, a button should be pressed to populate 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>:

Starting May 29, I will meet with 2-3 groups daily. During these meetings, the group will quickly show the teacher what they have implemented. The groups may ask the teacher questions about your website functionalities but do not ask questions about errors your code throws. The idea is to have a productive meeting and not spend time trying to solve code issues. For those errors, you may use Piazza. Post your questions and issues, and your peers or I will help. I will not respond to any question regarding code errors if I do not see it posted on Piazza. I encourage students to respond to their friends on Piazza if they know the possible solutions for their issues.

I plan to have 2 or 3 meetings (depending on time) with each group during the project development. The meetings will be graded (10 points each if we meet 3 times or 15 each if we only have 2 meetings). I will grade progress, discussion, and presentation. **(30 points)**

1. **2024-05-28 08:00 am (10 points)**
    - The first version of your design doc in the specified location (electronic)
    - Hardcopy (beginning of class)
2. **2024-06-03 08:00 am (12 points)**
    - 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
    - Branches setup for all teammates
    - API mechanism in place
    - Working database (API data should be stored in the database)
3. **2024-06-10 08:00 am (8 points)**
    - Requirements in place
    - Summary on readme
    - Launch codes in place
4. **2024-06-12 10:00 pm (25 points)**
    - The project manager must email the teacher with the subject **Final Project - Group Name**. Include the repository GitHub link. Attach these files: design.pdf, devlog.txt, README.md. CC'ed all the teammates.
5. **2024-06-12 10:00 pm (10 points)**
    - All groups must make a video (10-15 minutes) demonstrating their website functionalities, briefly explaining the code, and adding anything else you would like. All members of the group should talk during the presentation. I will send a form so students can share the location of their videos.
6. **2024-06-12 10:00 pm (5 points)**
    - Complete the evaluation form.

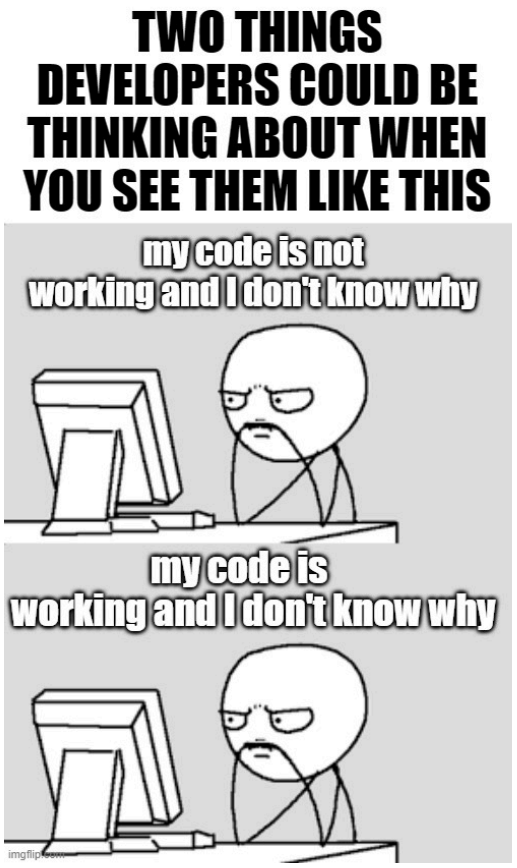
Reminder:

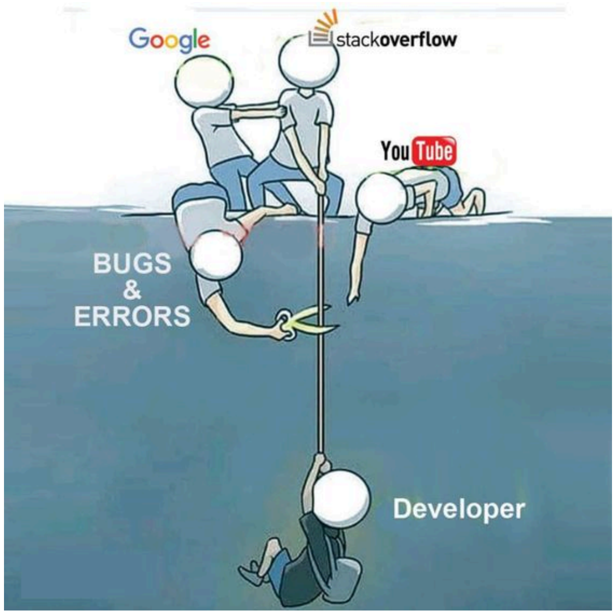
If you have questions during the development of this project, please ask the teacher. Do not wait until the last minute to ask questions.

If you are having code issues, try to research them. Then, try to **POST IT ON PIAZZA**. Your peers or teacher will help.

HAVE FUN & GOOD LUCK!!!

Frontend Dev
JSON
Backend Dev
imgflip.com


TWO THINGS DEVELOPERS COULD BE THINKING ABOUT WHEN YOU SEE THEM LIKE THIS
my code is not working and I don't know why
my code is working and I don't know why


Google
stackoverflow
YouTube
BUGS & ERRORS
Developer