Sprint 1: Project Proposal and UI Design

CS 4640

Project Description

Due: February 8, 2022 at 11pm

Purpose: Planning for and getting started with your project.

Part 1 - Project Proposal

Over the semester, you will learn how to use both client facing and backend technologies to make a web application / website. You will work on each homework with a partner, and over the course of the semester, will build a dynamic web application piece by piece. To maintain some cohesion between each assignment, and leave you with a final, portfolio-worthy deliverable, you will decide on a project concept. You and your partner will define your own project based on something that is interesting to your team.

You should first find a partner. If you are looking for a partner, please post on the Piazza thread for finding a partner.

Note: For this assignment, no technical implementation is required, unless you would like to get feedback on what / how you plan to implement. General idea is sufficient. Please include enough information about your project — the nature of your project, what you want to do, what you try to achieve, how you will achieve, etc. The purpose of this assignment is to get you to start. You may revisit, modify, add, or remove any part of your project description as you proceed / implement. The more detail you include in this assignment, the more feedback and comment I can be provided.

Reminder: You may not use the same project as another course or a project you have completed previously to satisfy the requirements of this course project.

Project requirements

1. It must include dynamic behavior, where the front end responds to user input events or web service and updates the interface accordingly.
2. It must include at least 3 functionalities (or scenarios), providing services to the users.
    - For each functionality (or scenario), be sure to describe its purpose, what it does, how it is used, and what the users can expect from using it. Later, you will implement each of these functionalities (or scenarios).
3. It must include logic that will execute both (i) client side in a web browser, and (2) backend component on a web server.
    - Examples: logic that executes on a client side
        - Validate the form data (client-side input validation)
        - Auto complete some information
        - Auto correct typos or misspelling words
        - Reformat the form data entries (textual, tabular, and graphical formats)
        - Filter search results
        - Sort the search results
        - Build a visualization (textual, tabular, or graphical) from a 3rd-party API
    - Examples: logic that executes on a server side
        - Validate the form data (server-side input validation)
        - Add / update / delete / retrieve data (from files or database)
        - Produce HTML documents; for example, reports or confirmation pages
        - Perform business logic; for example, calculate tax, compute grade or GPA
        - Handle HTTP requests from CURL command or URL rewriting

(HTTP requests that bypass the application's interface or client-side input validation, and thus may break the execution flow of the app). *Note: Bypassing the application's interface increases vulnerability and may lead to unauthorized access. If your project involves sensitive information, a thorough server-side input valiation is necessary.*

4. It must support multiple users (i.e., maintain states of the application such that multiple users can access the application simultaneously and their sessions do not overlap or interfere with each other).
5. It must support multiple sessions, allowing returning users to access / retrieve / manipulate their existing information or configuration (i.e., must include data persistence using a database).
6. The frontend and the backend must have an asynchronous component (with JSON):
    - The frontend (JavaScript) sends an asynchronous request to the backend (PHP).
    - The backend (PHP) processes the request and produces a response.
    - The backend (PHP) returns a response to the frontend.
    - The frontend appropriately uses the response from the backend (PHP) in some way.
    - *Note: It may seem unclear at this point how the frontend and the PHP backend components interact. This course will help you see how to properly implement and connect the frontend and the backend components.*
7. Overall, your app must be a single unit. It must not be disconnected. All parts / components must be properly connected.
    - That is, once a user enters your app through a web browser, there is an appropriate navigation system and/or an execution flow allowing a user to achieve her/his goal(s) without (re)visiting

another part of your app manually (for example, by entering another URL in a web browser's address bar).

Guidelines on what we expect to make a good project:

Your project should be something you and your partner can implement in a semester

- It need not be a feature-complete, fully functioning application, but may rather be a prototype that focuses on a few of the most interesting aspects of the application but does not implement or provide a full range of functionality (for example, a drawing application might only let the user draw, move, and resize lines).

Some project ideas

These project ideas are just to get you started. Please do not let them limit the possibilities; your project should be fun, something that you're passionate about and interested in, and excited to see built.

- Educational resource management system (recommended — may be used by faculty)
  - Allow users to register, login, and maintain account information
  - Allow users to view existing resources
  - Allow users to upload or download resources
  - Allow users to search or filter resources using a variety of different categories
  - Allow users to collaborate or provide comment
  - Provide sorting options
- Research project database and management system (recommended — may be used by faculty)

    - Allow users to register, login, and maintain account information
    - Allow users to view existing projects
    - Allow faculty to add, update, remove projects and number of student researchers needed
    - Allow students to search or filter projects using a variety of different categories
    - Allow users to collaborate or provide comment
    - Provide sorting options
- Self-paced practice system(recommended — may be used by faculty)
    - Allow users (educators and learners) to register, login, and maintain account information
    - Allow users (educators) to upload class rosters so that problem sets may be assigned to a particular group of learners
    - Allow users (educators) to update or delete his/her class rosters
    - Allow users (educators and learners) to view existing problem sets
    - Allow users (educators) to create, update, or delete his/her problem sets
    - Allow users (educators) to collaborate, share, or reuse problem sets.
    - Allow users (learners) to practice (assigned or unassigned) problem sets
    - Allow users (educators) to keep track of learners' progress
    - Provide an intervention mechanism such as sending emails to users (educators and/or learners) automatically
- Jeopardy game (recommended — may be used by faculty)
    - Allow users to register, login, and maintain account information
    - Allow users to create questions; specify the types of questions (such as multipole choice, true/false, or short answer); enter answers, depending on the question types
    - Allow users to maintain (add, update, delete) their questions and

answers
    - Allow users to select their questions to create jeopardy games and assign scores to the questions
    - Allow users to maintain the game information (such as titles, instructions, list of questions and scores, game layouts)
    - Allow users to play the existing games and maintain scores
- Personalized news visualization
    - Allow users to register, login, and maintain account information
    - Aggregate a large number of different types of news articles
    - Allow users to maintain their preferences, allowing customization
    - Allow users to view the articles; the latest happenings from around the world; an overview of current news
    - Allow users to search or filter the articles using a variety of different categories
    - Provide visualization of the news
- Invitation system
    - Allow users to register, login, and maintain account information
    - Allow users to create events (with information such as name, date, time, location, host)
    - Allow users to maintain (add, update, delete) their events
    - Allow users to send invitation to other users
    - Allow users to respond “attend”, “not attend”, or “maybe” on other users’ events
    - Allow users to leave messages to the hosts
    - Provide sorting options
- Review system (things to be reviewed such as movies, products, restaurants)
    - Allow users to register, login, and maintain account information
    - Allow users to maintain (add, update, delete) information about the things to be reviewed

    - Allow users to view existing list of things
    - Allow users to search or filter things using a variety of different categories
    - Allow users to rate and provide comment about things
    - Provide sorting options
- Scheduling system
    - Allow users to register, login, and maintain account information
    - Allow users to maintain (add, update, delete) their arrangements (such as tasks, events, appointments or meetings)
    - Allow users to view existing arrangements (calendar or agenda)
    - Allow users to share schedule
    - Allow users to search or filter arrangements using a variety of different categories
    - Allow users to drag and drop arrangements

Optional Functionality

In addition to the above requirements, you may make use of any APIs or web services to enhance your project. Please be sure to document and justify your design decision.

Example web service APIs:

- 22,353 APIs on programmable web - www.programmableweb.com/apis/directory
- Marvel - developer.marvel.com/
- Google cloud entity recognition - cloud.google.com/natural-language/
- Twitter - dev.twitter.com/overview/api
- Google Maps - developers.google.com/maps/documentation/javascript/
- NFL play by play day - nflsavant.com/about.php
- Bing news API - www.microsoft.com/cognitive-services/en-us/bing-

news-search-api
- Yahoo - Weather, photos, finance, etc. - developer.yahoo.com/everything.html
- Instagram API - www.instagram.com/developer/
- GitHub API - developer.github.com/v3/
- SNAC Cooperative - snaccooperative.org/api_help

Part 2 - User Interface Design

Design at least three different screens or views (with respect to the three functionalities or scenarios above) of your project.

Each screen must adequately demonstrate its purpose and functionality. You should include all necessary elements and justify your design decision. Note: you may revisit and update your UI design while implementing the components.

To expand your experience and help you to move toward careers in web profession, you should try to create a digital illustration of your interfaces. You may use any software to draw your screens. There are several good, easy to use web-based software, standalone software, and wireframes (some are free, some offer free trail) that you can use to draw user interfaces:

- Balsamiq
- MockFlow
- Pencil Project
- Framebox

Alternatively, Visio, PowerPoint, Paint / Preview, and Adobe Photoshop can also be useful.

If you draw your screens by hand, please be sure that your drawings are legible; take screenshots of your drawings and embed them as part of your proposal document.

If you draw your screen using any software, please be sure to embed your screens as part of your proposal document.

Grading Rubric

For this sprint, you will submit a brief report in two parts:

1. (20 points) Project Proposal
    - (10 points) Clearly explain your project concept
    - (10 points) Satisfy the project requirements
        - Clearly describe how your project concept satisfies the project requirements or specify the parts of your concept that satisfy the requirements
2. (20 points) User Interface Design
    - (15 points) Include at least 3 screens (5 pts each)
    - (5 points) Clearly explain your design, justify your design decision, discuss how your design decision affects the usability

- (-4 points) All team member names are not included when submitting
- (-4 points) Submitting the report in Word (i.e., **not** submitting a PDF)

Submission

Upload your report as a **PDF** to the Project Proposal assignment on Gradescope. Name your file with all team members' computing IDs, such as *jh2jf-mst3k-proposal.pdf*. Make sure you connect your partner to your group on Gradescope so that everyone receives credit.