

Course Project: Shared Workspace Web App

Coworking is a work style where different individuals or teams share a working environment. If you are a self-employed or a working-from-home professional, you would love to have the option to rent a meeting room, or a desk in an office equipped with high-speed internet, printers, copiers, stationaries, and office kitchen. As there is an increase in the number of coworkers and the number of sharable workspaces, we are looking to develop an application to connect both parties. The objective is very similar to how Uber connects riders with drivers and Airbnb connects tenants with landlords.

For this project, you must use the frameworks we cover in class. If you want to use any other framework/technology, you should ask the permission from the instructor.

1. Design and implement a standard web application for the given shared workspace project. The web application wireframe should have landing page, header and footer. For more functionalities refer to the user stories to figure out what pages do you need and what controls should each page has. Whenever you want to store any information you can use array or Js object at this level.

The following is a guideline:

User Story 1: Add a web page with controls to take the user information from keyboard and an Add button. On click on the Add button, adds users to the user object or array.

User Stories 2 and 3: similarly add web pages for the owner user to specify the property and workspace information and click a button to call the corresponding web service/ functionalities that adds them to the property/workspace object or array dataset.

User Stories 4 to 8:

Add a web page with controls to search and display the properties and their workspaces. Add a login page. If the login is successful, based on the user's role, modify the previous page as follows:

- For the owner user, filter the page to show only the owner's properties and workspaces. Also, show controls that the user can use to sort, update and delete any of the items.
- For the coworker user, show controls for the user to search and to sort properties using different criteria.

All pages should make requests to your locally defined array object or JS Object application.

Minimum Viable Product (MVP)

The MVP is the smallest conceivable list of features that fulfill the primary business goal of the software product. One way to summarize a feature is a user story-- a short sentence describing the feature from the perspective of the user. User stories frequently take the form: "As a <type of user>, I can <take some action> so I can <some reason>." You can start with the following user stories as a backlog for your MVP.

From your web application a user can do the following activities (Given user stories)

1. As a user, I can sign up for an account and provide my information (name, phone, and email). I can choose my role as an owner or a coworker.
2. As an owner, I can list a property with its address, neighborhood, square feet, whether it has a parking garage, and whether it is reachable by public transportation.
3. As an owner, I can select one of my properties and list workspaces for rent. Workspaces could be meeting rooms, private office rooms, or desks in an open work area. For each workspace, I can specify how many individuals it can seat, whether smoking is allowed or not, availability date, lease term (day, week, or month), and price.
4. As an owner, I can modify the data for any of my properties or any of my workspaces.
5. As an owner, I can delist any of my properties or any of my workspaces.
6. As a coworker, I can search for workspaces by address, neighborhood, square feet, with/without parking, with/without public transportation, number of individuals it can seat, with/without smoking, availability date, lease term, or price.
7. As a coworker, I can select a workspace and view its details.
8. As a coworker, I can get the contact information of a workspace's owner.

Bonus Features

Adding the following features will give you the bonus marks.

1. As an owner, I can add photos to any of my properties or workspaces.
2. As a coworker, I can rate a workspace on a one-to-five-scale. The workspace's listing should include the average rating.

3. As an owner, I can rate a coworker on a one-to-five-scale. The coworker's profile should include the average rating.
4. As a coworker, I can sort the workspaces' search-results by the average ratings, availability date, or price.
5. As a coworker, I can write a review about a workspace. The workspace's listing should include these reviews.

Submission Requirements

You will submit the following documents for your project.

1. ReadMe file that describe how to execute the code, initial setup etc.
2. All the code files including the packages that you have used.
3. You will also submit a video describing the functionalities of your project.

Submit everything in one zip file over D2L.

This project worth **30%** of your total mark.

Note:

If you want to use any other project, you are welcome to do that. However, you should first take permission from the instructor. I must mention that TODOAPP is not a valid project.