

CPSC 431-01 Spring 2024 – Final Project

Requirements Specifications

Instructor: Steven Cooper

Updated 2024/04/15

Final Project: Create a Simple Web Application that uses a Data Source

Review the following summaries and then use the detailed specifications to create your project. Grading rubrics will be provided on Canvas.

Option 1: To-Do List

For this project the application is a to-do or task list. The capabilities will specify that the application should allow users to create a list of text items which they can then mark/unmark as completed. Data will persist by using a database. Users can create multiple lists and modify or delete them as well.

Option 2: Appointment Manager

For this project the application is a virtual appointment book or event calendar. The capabilities will specify that the application should allow the user to select a time and date and create an event with a description for that specific time slot. The application will not allow users to create overlapping events and should display existing events so users can identify what times are available. Users can create, view and delete events.

Deliverables (What to Turn In)

To turn in your assignment and receive credit, you will need to perform the following tasks:

1. Make sure your application (Back-end PHP and Front-end HTML) are in one folder that works from your *htdocs* folder in Xampp. **Archive this folder into a .ZIP file** and upload it to

Make sure your .ZIP file uses this naming convention: **431Final_CooperSteven.zip**
(Obviously replace my name with yours)

2. If you used a MySQL database then export it to a .SQL file **with the “Create Database” option checked.**

The name should follow this pattern: **CooperSteven_431Final.SQL** or **CooperSteven_431Final.json**.

Make sure the DB name follows this pattern: **431Final_CooperSteven** in phpMyAdmin. You can manually change this in the .SQL file if you named it incorrectly.

Your SQL statement should create a database with table schema and stored procedures. You may export only schema and not use any dummy/premade data if you like. Your app should be able to create new records anyway.

3. If you decided to use a simple **.json** file and write directly to the file, you can simply let it be a part of your **.zip** file and you do not need to submit 2 separate files.
4. Upload your **.zip** file (and your **.sql** file if you used MySQL) to Canvas.

Global Requirements

All applications must conform to these requirements regardless of which option is selected.

Constraints

Requirement	Details
XAMPP 8.2 Environment	Application must run in XAMPP 8.2. Any additional software must be provided.
Persistent Data (Server-Side)	Data can be stored in a Database or as JSON but must be stored outside of the client's browser (i.e. not in local storage)
HTML/CSS Interface	The GUI should make use of standard HTML input elements (inputs, buttons, hyperlinks) for user interactivity.
WCAG 2.0 AA Compliance	Make a reasonable attempt to follow the AA color contrast requirements for accessibility. https://webaim.org/resources/contrastchecker/

Capabilities

There are no global capabilities.

Terms and Definitions

These terms apply to all project variants.

Term	Definition
User	The person using the application.
API	The code that communicates between the data source and the client.
Data Source	Where the data is stored (MySQL DB or a simple JSON file)
Client	The software on the user's machine, the browser (Chrome, Firefox, Edge, Etc.)
Client-Side Code	The JavaScript code running in the user's browser.
Server	The Apache server which processes HTTP requests and runs PHP Code
Asc/Ascending	A sort order where the smallest or newest values are displayed first. [0-9, A-Z]
Desc/Descending	A sort order where the largest or oldest values are displayed first. [9-0, Z-A]
Loaded	App loaded means whenever the web app page is initially visited by the user.

To-Do List – Requirements and Specifications

Use the following capabilities and constraints to create a To-Do List web application.

Constraints

Each requirement is a constraint, rule or environment specification your application must follow.

Requirement	Details
Confirm List Delete	Present a confirmation page or window and require users to select “Yes” or “Okay” before executing a DELETE request to remove a record from the DB.
Attributes for “List” Record:	idx (Index), name (String), created (Datetime)
Attributes for “List Item” Record:	idx (index), text (string), checked (Boolean), list_idx (FK to list table), created (datetime)
List Item Sort	List items are sorted by dimensions: checked and then created, descending. Checked items should always appear first and then all items are sorted oldest to newest.
Default List Sort	Lists are sorted by creation date (ascending) when the app is first loaded.

Capabilities

Each requirement represents an action that the user should be able to perform using the interface.

Requirement	Details
View summaries of all Lists (Read)	User can view all created lists, only the list name needs to be visible.
View a List (Read)	User can view a specific list and all of the list items inside.
Create a List	User can create a new list by providing a name for it.
Add/Remove List Items (Update)	User can add or remove list items to a list with the UI.
Check/Uncheck List Items (Update)	User can check/uncheck a list item and this status persists in the database.
Rename a List (Update)	User can change the name of a list.
Delete a List	User can permanently delete a list and all of its List Items .
Sort List summaries	User can sort the display of lists by either “name” or “created”, asc. or desc.

Terms and Definitions

These terms are specific to this project variant.

Term	Definition
List	A collection of List Items.
List Item	A string that can be stored in a list.

Notes

- You can determine text size limits. A list item probably shouldn't be extremely long, maybe 1-2 sentences max.
- A MySQL DB is recommended for this project as a JSON file would be extremely difficult to modify since it is not a relational data store.

Appointment Manager – Requirements and Specifications

Use the following capabilities and constraints to create an Appointment Manager web application.

Constraints

Each requirement is a constraint, rule or environment specification your application must follow.

Requirement	Details
Confirm Delete	Present a confirmation page or window and require users to select “Yes” or “Okay” before executing a DELETE request to remove a List record from the DB.
No Overlapping Dates	Users cannot create 2 appointments on the same dates with an overlapping time range
Past Appointment Warning	Provide a visual/text warning if the user is creating an appointment where the date and time is in the past instead of the future.
Appointment Details	Collect this info for appointments: Title, Date, Start Time, End Time, Description

Capabilities

Each requirement represents an action that the user should be able to perform using the interface.

Requirement	Details
Create Appointment	User can create a new appointment for a specific date and time range.
View Appointments	User can view all of their appointments
Update Appointment	User can modify the details of a particular appointment
Delete Appointment	User can delete an appointment permanently
Filter Appointments	User can filter appointments they are viewing by “Upcoming” & “Past”

Terms and Definitions

These terms are specific to this project variant.

Term	Definition
Upcoming Appointments	Appointments that start in the future
Past Appointments	Appointments that happened in the past
Details	The title, description, date, start time, and end time for the appointment

Notes

- SQL is all but required for this, it would be near-impossible to make this work with a JSON file.
- It's probably easier to compare dates in PHP or SQL than JavaScript. You can write a SQL query just to compare dates and use PHP to get the results though doing this in PHP is recommended.
- Data types are up to you, you should really only need text (VARCHAR) and Datetime.
- Filtering can be done in JavaScript alone but you may want to do it in SQL with a procedure/query because it could be much easier.