

Melon Tasting Reservation Scheduler

Summary:

We want to build a simple service to help users make reservations to go to a fancy melon tasting!

We offer coverage 24/7/365 (including weekends and holidays) but only 1 user can book an appointment on a given day and time.

Assignment

Your task is to build a web application that users can use to search for and book reservations.

A user should be able to “log in” (by entering their username), pick a date, choose an optional time range, and then be shown all the available reservations that meet the criteria.

They can then book an appointment of their choice. If there are no reservations available, you should display a message indicating that.

There should also be a page which shows all reservations for a given user.

The service has these requirements:

- all reservations must start and end on the hour or half hour
- all reservations are exactly 30 minutes long
- a user can only have 1 reservation on a calendar date (*#tooMuchMelon*)

If these conditions cannot be met (for example the user has already booked an appointment on the chosen date), show an error message indicating that.

Your application frontend should have the following pages:

- 1) Simple login screen (no authentication needed; more on this in the details section)
- 2) Page to search for appointments (see following page for an example of this)
- 3) Results page to view the results of the appointment search (allowing users to book any of those appointments and showing them an error if none are found)
- 4) Page to view all the scheduled appointments for the current user (you can optionally build canceling or editing of appointments but this is not required)

*Sample appointment search page (yours can be designed as you see fit and the input fields can be regular HTML 5 date and time input elements if you would like - **You do not need to build a calendar or time input yourself!**)*

○ ○ ○

Melon Reservation Scheduler

Enter the date of the reservation you would like

< February 2017 >

Mo	Tu	We	Th	Fr	Sa	Su
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28

25 Aug 2021



Enter an optional time range and we will only show appointments in that range

Start time

End time

Search

Details

We are intentionally not dictating what technical requirements you should use. You will need some kind of database (to persist the user and appointment information), a backend server, and frontend.

If you are familiar with a date library, such as Python's `datetime` or Javascript's `moment` or `date-fns`, feel free to use that. You can assume all timestamps are in the same timezone (no timezone handling required).

For the frontend, you can use plain HTML (by itself or with "Vanilla" Javascript) or a library such as React to build a single page app.

If you are less familiar with these libraries, no problem! Use what you are comfortable with.

The application should be easy to use but CSS and styling should not be a priority. Feel free to use a front-end framework or UI library such as Bootstrap or Material UI to help with this, but built-in HTML elements are totally fine as well. You can assume the user will have a modern web browser with no major browser compatibility issues.

You do not need to handle user sign up – you can create one or more test users in your database. If you write a seed file, please include it in the final submission. If you need sample user data to get started, you can use <https://jsonplaceholder.typicode.com/users> or a tool like Faker.

There is no need to implement login using passwords – you can assume a user can just log in using their username or email. If you do want to build password authentication as a bonus feature, you can use a library if that is helpful.

Your server should preserve data across restarts (in other words you should persist information in some kind of database or other permanent storage space).

Feel free to add any extra features, validations, or error handling if you have time. If you do add extra things, do make sure you mention them in your README.

Submission

Please push your code to Github (can be a public repository). To submit, you can post a link to the finished Github repository in your personal Discord channel.

In addition to your source code, README, and dependencies files (like requirements.txt or package.json), you may include other files like helper scripts or unit tests. However, these additional files are NOT required.

README

Please include a README with anything you would like us to know (why you chose the technologies used, how you structured your database, any trade offs you made and why, any additional pieces you added, etc.)

The README does not need to be exhaustive or long – we are more interested to learn about your decision making process and anything interesting you learned or want to highlight.

Deployment

You should deploy your finished site and send us a link.

We recommend a service like [Render](#) for ease of deployment. You do not need a custom domain and using a free tier should be sufficient.

Time

This exercise is intended to take less than a day (4-6 hours). Please spend no more than 6 hours total, including finalizing and sending your submission.

If you don't get around to everything you'd like to do, describe in your README or add TODO comments to indicate what you'd like to add/change given more time.

But do not spend more than 6 hours – at that point please submit what you have done.