



ENG1003 - Engineering mobile apps - S1 2021

[Dashboard](#) / [My units](#) / [ENG1003_S1_2021](#) / [Assignment 1a \(Individual\)](#) / [Assignment 1a Instructions](#)

**Information**

Teaching for the unit has now been completed. Learning Captures are no longer available. The Lectures and Unit Coordinators will no longer be accessing or monitoring this site. This Moodle site will remain available to you as resource until you have completed your studies at Monash. Please note the materials presented here were accurate as of time of Teaching and may be superseded at any time.



Assignment 1a Instructions



2. Tasks

You will be building a simple single page web app that will run and allow users to book a specific room between the hours of 8AM and 6PM, in 1 hour blocks. Before you begin work on this assignment, make sure you read and study the entire set of tasks to understand how the different components interact with one another.

You will be creating one HTML and one JS file. Please name them as follows:

- index.html
- main.js

Your files should follow the naming convention and folder structure outlined under the 'Submission and Late Penalties' section

Study the data provided below

```
let bookingData = [,,,,,,,,,
  {
    time: "08:00",
    reason: "",
    label: "",
    booked: false
  },{
    time: "09:00",
    reason: "",
    label: "",
    booked: false
  },{
    time: "10:00",
    reason: "",
    label: "",
    booked: false
  },{
    time: "11:00",
    reason: "",
    label: "",
    booked: false
  },{
    time: "12:00",
    reason: "",
    label: "",
    booked: false
  },{
    time: "13:00",
    reason: "",
    label: "",
    booked: false
  },{
    time: "14:00",
    reason: "",
    label: "",
    booked: false
  },{
    time: "15:00",
    reason: "",
    label: "",
    booked: false
  },{
    time: "16:00",
    reason: "",
```



```
    label: "",
    booked: false
  }, {
    time: "17:00",
    reason: "",
    label: "",
    booked: false
  }
];
```

Copyright Monash University 2018. Copyright Warning You are logged in as Jin Chun Tan (Log out)

This data stores the booking data for a given room in the system. Each item in the array contains an object with the information of the booking status of that time slot. The timeslot of the object corresponds to the index in the array. This information consists of:

- time: the starting time in 24-hour time of the time slot
- reason: the reason for this booking, which is stored internally and not shown to users
- label: the publicly shown description for the booking
- booked: a boolean flag to indicate if the room is currently booked or not

Task 1

Write a function named **bookRoom** that takes the parameters **time**, **reason** and **label**. This function is responsible for updating the main booking data for the room. It should update the reason, label, and booked status for the time specified.

Task 2

Write a function named **checkRoomBooked** that takes the parameter **time**. This function is responsible for checking the booking data to see if the room is booked at the time provided. It should return true if the room is booked, and false if the room is available.

Task 3

Write a function named **clearRoomBookings**. This function will 'reset' the bookings by resetting the appropriate properties for each time slot to their default state. It should iterate through the relevant time slot objects and set the relevant properties to the default state (like in the initial data provided above).

Task 4

Create a HTML page that meets the following specifications. A screenshot of the final product is shown below.

Room Booking Page

Time:

Start Time:

Booking Reason:

Booking Text to be shown:

Room Booking Status

Page Title: "Assignment 1a"

Page Contents (body):

- div
 - h1: "Room Booking Page"
 - h4: "Time:"
 - span (id: timeNow)
- div
 - label: "Start Time:" (for: inputTime)
 - select (id: inputTime)
 - option (see description below)



- option (see description below)
- div
 - label: "Booking Reason:" (for: inputReason)
 - input (id: inputReason)
- div
 - label: "Booking text to be shown:" (for: inputLabel)
 - input (id: inputLabel)
- div
 - button: "Book Room"
- div
 - h3: "Room Booking Status"
 - div (id: output)
- button: "Clear All Bookings"

You should use the **text** type for input fields.

In the options for the dropdown list (id: inputTime), you should create items that show the following times

- (empty)
- 8am
- 9am
- 10am
- 11am
- 12pm
- 1pm
- 2pm
- 3pm
- 4pm
- 5pm

It is suggested that you make the value of the option the 24-hour number (no leading 0) of the time (i.e. 15 for 3pm). This would look like the screenshot below:



You can also include the following bit of **CSS** in your head element using a style element for spacing between each row of content (divs).

```
div {
    padding-bottom: 10px;
}
```

You should also link your JS file by including a script element inside your head element with the src attribute pointing to your file.

Task 5

Now we will be writing some functions to implement functionality between the DOM and the JS code already written.

Write a function named **updateDisplay**. This function is responsible for iterating over the *bookingData*, generating and displaying the necessary HTML code to show the current booking status of the room for the day on the page to the user.

You must clearly show the following:

- timeslot
- booking status (available / not available)
 - if not available, the booking label must be shown.

If the room is available at a given time slot, it should show "Available". If not, it should show "Not Available (label)" and show the booking label in the brackets.

You should use an output variable, and display the generated HTML, on the HTML page in the div with the id of **output**

You should use an output variable, and display the generated HTML on the HTML page in the div with the id of **output**.

You can choose to use a **p** element to contain each line, or use a **br** element to make a new line of text.

Here's a screenshot of the suggested output on the page, showing both the elements above and below the **output** div element.

Room Booking Status

08:00: Available

09:00: Available

10:00: Available

11:00: Not Available (ENGINE Meeting)

12:00: Available

13:00: Available

14:00: Available

15:00: Available

16:00: Available

17:00: Available

Clear All Bookings

Task 6

Write a function named **doBooking** that is responsible for interacting with the DOM to make the booking using the user input.

You must do some basic validation on the user input. This includes

- checking that the label and reason is not empty.
- checking that the time is selected.
- checking that the time selected is available for booking (by calling your Task 2 function).

If the input is not valid, you should show an appropriate error message to the user using **alert()**, and use **return** to end the function execution.

If all the input is valid, then you can make the booking by calling your Task 1 function.

Then, update the display by calling Task 5 function.

Here's some suggested pseudocode for this function:

1. Confirm with the user that they want to make this booking.
2. Create references for the user input elements (time, reason, label).
3. Create variables to hold the value of the user input from the references in (1).
4. Do some validation on the user input
 - a. If any of the inputs are invalid, show an appropriate error message and end execution.
5. Make the room booking.
6. Update the display on the page.

Finally, you need to go back to your HTML file and modify the "Book Room" button and add an **onclick** attribute to execute your function from this task.

Task 7

Write a function named **clearAllBookings** that is responsible for interacting with the DOM to clear all bookings for the day.

You should confirm with the user first that they want to clear all bookings.

To clear the bookings, call your Task 3 function, then update the display by calling Task 5 function.

Finally, you need to go back to your HTML file and modify the "Clear All Bookings" button and add an **onclick** attribute to execute your function from this task.

Task 8



Write a function named **updateDayTime** that will display the current time on the page for the user.

This function should show the current time (time only, no date) on the page in the span element with the id of **timeNow**.

Task 9

Finally, we can write some code that will run when the page loads so that all of the above functionality will function as intended.

At the bottom of your JS file, write the following:

1. Create a timer that will run every 1000ms that calls your function from Task 8. This will allow the 'live' updating of your clock on the page.
2. Call your function from Task 5 so that the room booking status is also shown when the page loads.

Then clean up your code and document your work. Make sure that you :

- have appropriate code Indentation and Code Brackets (per the style guide).
- have appropriate variable and function naming (per the style guide).
- have appropriate variable and function scoping (smallest possible).
- have appropriate function header documentation (name, purpose of function, parameter and return value description).
- start the JS file with `"use strict";`



◀ 12.2 Security and Privacy

Jump to...

Assignment 1a submission (Due W5 Thu 8PM) ►