Message to the candidate:

Thank you for taking the time to interview with us at Venturit! As part of the interview process, we'd like to evaluate your technical abilities to understand how you would fit into our engineering organization.

We'd like you to complete the following take-home coding exercise over the next 72 hours. *(If you need a few more hours, we're flexible - just don't take much more than that!)*

Please create a code repository in a setting that can be shared (such as GitHub, Gitlab, or Bitbucket). In that repository, write code to complete the following projects to the best of your ability.

When you've finished, please ensure that your code is uploaded to the repo, and that the repo will be visible to our interviewers, and one or two members of our engineering team. Then, email a link to the repo to your recruiter, who will ensure that the interviewers can review the code.

We'll complete the process by having you review your code with those members of our engineering team as part of your final round interview.

**We highly recommend you double-check that your repo is easily accessible, whether that's through making it public, providing an "anyone can view this" link, or, if no other options are available, attaching a zip archive of the code to an email.*

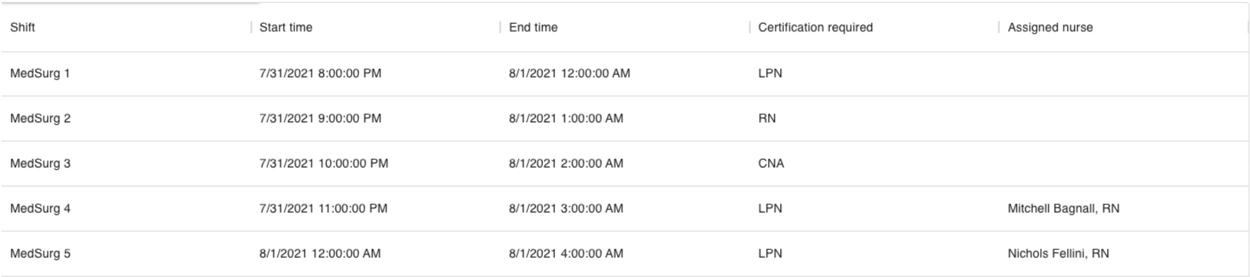
This project may be done with any combination of web/mobile languages and libraries that the candidate is comfortable with.

For this project you will be creating a simple webapp for tracking the nursing shift schedule at a care facility. We will provide you with a general UI wireframe for this webapp, as well as a mocked API server to provide data for your app. Please build this app as reasonably close to the specification as you can. If you can't finish everything, we'd prefer a solution that fully completes some of the parts rather than partially completing all of the parts. If you have extra time, we'll welcome any additional polishing you have in mind.

Part 1 - Data fetching and table building

- When the app loads, fetch the list of shifts from the server into your app's state.
- Also, fetch a list of nurses from the server and load that into your app's state.
- Using the shift and nurse data, render a table showing every shift in the schedule.
- The table should have the following columns:
    - Shift (the name of the shift)
    - Start time
    - End time
    - Certification required
    - Assigned nurse (the first name, last name, and qualification of the assigned nurse, if there is one)

Here is an example screenshot of how part 1 might look when you've finished. Depending on your choices of libraries and styling, yours might look different, and that's okay.



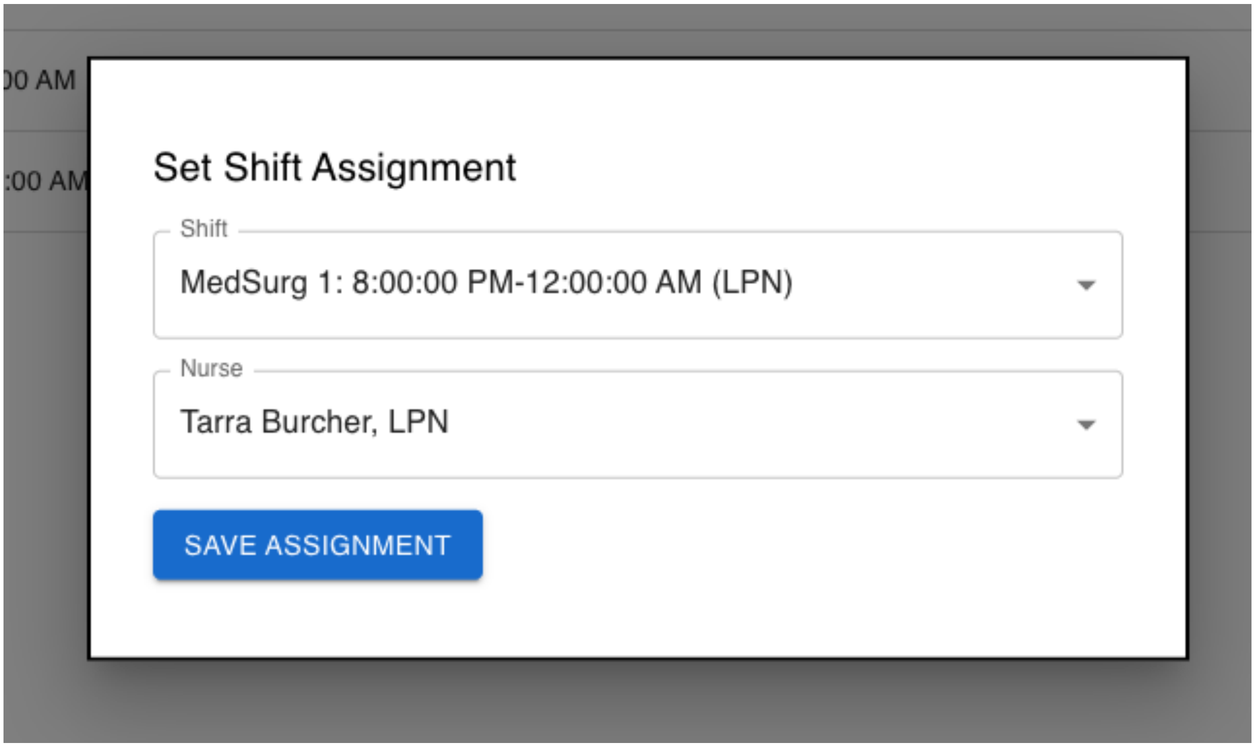
Shift	Start time	End time	Certification required	Assigned nurse
MedSurg 1	7/31/2021 8:00:00 PM	8/1/2021 12:00:00 AM	LPN	
MedSurg 2	7/31/2021 9:00:00 PM	8/1/2021 1:00:00 AM	RN	
MedSurg 3	7/31/2021 10:00:00 PM	8/1/2021 2:00:00 AM	CNA	
MedSurg 4	7/31/2021 11:00:00 PM	8/1/2021 3:00:00 AM	LPN	Mitchell Bagnall, RN
MedSurg 5	8/1/2021 12:00:00 AM	8/1/2021 4:00:00 AM	LPN	Nichols Fellini, RN

Part 2 - Form building

- Add a button to the app, located above the data table, with the text "Set Shift Assignment".
- When the button is clicked, it should open a modal with the following components inside it:
    - A header with the text "Set Shift Assignment".
    - A selection dropdown to pick one of the shifts from the overall list. The input text should display the name of the shift at a minimum.
    - A selection dropdown to pick one of the nurses from the overall list. The input text should display the first name, last name, and qualification of the assigned nurse.
    - A button with the text "Save Assignment".

- If the Save Assignment button is clicked, or if the user interacts in a way to close the modal (such as clicking outside it or clicking an X button, depending on your UI library), the modal should close.

Here’s an example screenshot of how part 2 might look when you’ve finished.



Set Shift Assignment

Shift: MedSurg 1: 8:00:00 PM-12:00:00 AM (LPN)

Nurse: Tarra Burcher, LPN

Part 3 - Saving

- Make it so that the state of the selected shift and selected nurse are reset whenever the form closes, with a default of having nothing selected.
- Make it so that the Save Assignment button is disabled unless a shift is selected.
- Update the Save Assignment button to trigger a save request against the server when it is clicked.
- In addition to sending the save data to the server, make it so that your app updates its internal state of shift-nurse assignments when a save happens.
    - We’ll determine if this works correctly by seeing if the table updates with the newly-assigned nurse after a save happens.

Part 4 - Form validation

- Make it so that the form will display an error message if the nurse is already working a shift that overlaps the time for the selected shift.

  - For one example, you can’t work a shift from 2:00-6:00 if you’re working an existing shift from 3:00-7:00. However, working from 2:00-6:00 and 6:00-10:00 is allowed.
- Make it so that the form will display an error message if the nurse is not qualified to work the selected shift.
  - To decide if a nurse is or isn’t qualified, see our definition in the box below.
- Make it so that while the form is displaying an error message, the save button should be disabled.

Here’s our simple definition of nurse qualifications for this project:

A nurse at the facility will have one of three levels of certification:

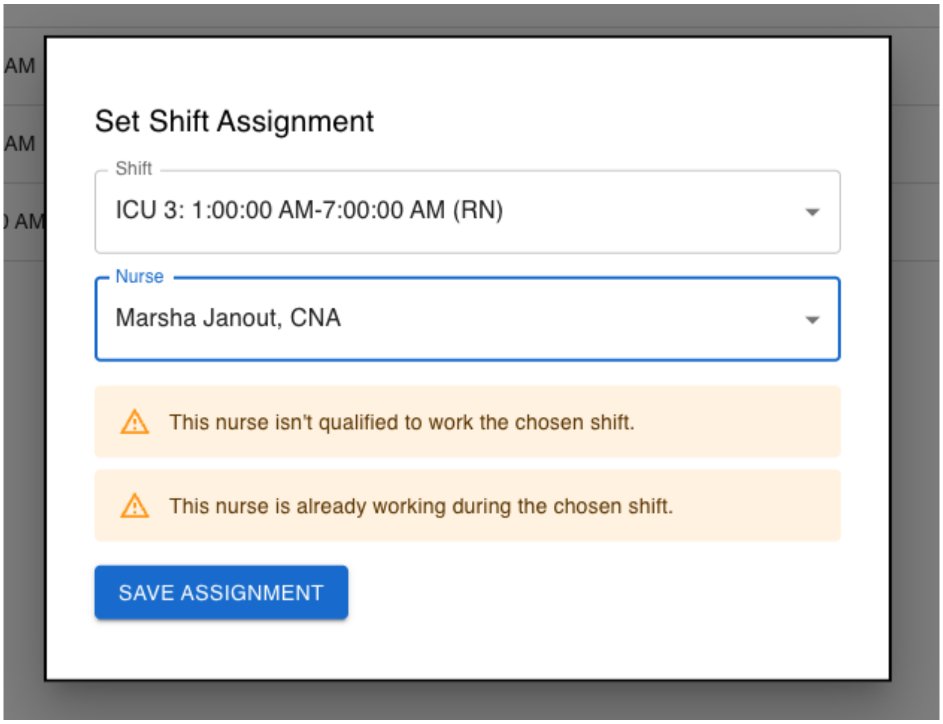
- Certified Nursing Assistant (CNA)
- Licensed Practical Nurse (LPN)
- Registered Nurse (RN)

A CNA can only work CNA shifts.

An LPN can work CNA or LPN shifts.

An RN can work CNA, LPN, or RN shifts.

Here’s an example screenshot of how part 4 might look when you’ve finished.



Set Shift Assignment

Shift: ICU 3: 1:00:00 AM-7:00:00 AM (RN)

Nurse: Marsha Janout, CNA

⚠ This nurse isn't qualified to work the chosen shift.

⚠ This nurse is already working during the chosen shift.

API

GET /shifts
Returns a JSON object with an array of shifts, each shift has an id, a start and end UTC datetime, a nurse ID (or null), and a qualification level (either CNA, LPN, or RN).

GET /nurse
Returns a JSON object with an array of nurses, each nurse has an id, a first and last name, a username, and a qualification level.

PUT /shifts/${shift_id}
A save routine that, given a shift ID in the route, and a nurse ID in the body, simulates saving the nurse to the given shift (this could either be a dumb auto-succeed passthrough or run some server-side validation that the nurse is qualified and capable of working that shift)

Supporting files for this exercise

`shift_list.json` - a JSON file with the list of shifts for this facility.

`nurse_list.json` - a JSON file with the list of nurses for this facility.

`server.js` - a nodeJS server that implements the API listed above. It delivers the data from the two JSON files for you to use with your app.

Server usage instructions

Save `shift_list.json`, `nurse_list.json`, and `server.js` into a directory together.
Make sure you have nodeJS installed and accessible on your machine.
Run the server with the command `node server.js`. It'll run on your localhost port 9001.