Instructions for IS24 – Full Stack Developer

Build a simple Web application that tracks and manages boats as described in the story provided below.

In order to move forward to an interview, you must pass a minimum of **two** of the following categories in your coding challenge submission:

- Front-end
- Back-end *
- Quality Assurance (QA)
- CI/CD (an opportunity to score additional points if the deliverables in this category are included in the submission)

* ***you MUST have a backend component completed and functional***

Take notes of your decisions, observations, and assumptions and include them in the submission. If you move on to the interview stage of the competition, you will be asked to deliver a short presentation (5-10 minutes) describing your decisions and thought process for designing the application.

What to submit

- A link to a publicly accessible cloud source control repository (e.g. GitHub, GitLab) with the complete source code for the application and any accompanying notes. The evaluation will be based on the code in the **master branch**. Any code commits pushed to the repository after the deadline will not be considered for evaluation
- Link(s) to the running version of your application which are publicly accessible on the web

Back-end Component

Build a Web service using a server-side framework (e.g. Express, Django, .NET, Go) and host it on a cloud platform service (e.g. Heroku, AWS, Azure, Google Cloud Platform).

User Authentication/Authorization is not required for this code challenge.

If your app does not include a front-end component, include a list of the backend API endpoints in the README file.

Front-end Assessment

Build a Web App **using React JavaScript framework** and host it on a cloud platform service (e.g. Heroku, AWS, Azure, Google Cloud Platform).

Quality Assurance Assessment

Write executable tests (e.g. Unit, API, Functional) for the Web App.

Define assumptions and detailed test/use cases on the story and/or write some functional tests against your boat tracker.

DevOps Assessment

Implement a deployment pipeline (e.g. GitHub Actions, CircleCI, Argo, etc) which checks out, builds, tests, configures, and deploys your application to a cloud hosting service (e.g. Heroku, AWS, Azure, Google Cloud Platform) or your own hosting environment.

Context

Fishfry Tours is a small salmon guiding outfit based out of Cascadia, British Columbia - along the coast. They run a seasonal guided sport fishing tour of some of the more hidden inlets of Coastal British Columbia. In total they have 8 sport fishing boats with 12 guides. At any given time there are at least 4 or 5 boats out in the waters. Sometimes the boats will meet each other to exchange gear and fuel for longer days at sea.

The control office maintains a kanban-like control chart on a white board which describes the state of each boat. Some of the swimlanes are 'Docked, Outbound to Sea, Inbound to Harbor, Maintenance'

The Boat Guides have expressed interest in having the control chart accessible online through their mobile phones (whenever there is service). Sometimes radio contact to other boats is not possible and using satellite services are too expensive to maintain constant communication.

One of the boat guides is a web developer during the off season and offered to build the app. He insists on building it using an Agile Approach.

The boat guides have varying computer skills. They mainly want to see the status of all the guide boats in the area at a glance and be able to move their cards into different swimlanes as needed.

Personas

Bob is 26 and has been a guide for most of his life. He has a dog named Wilfred that has spent more time at sea than on land. Bob is not very technically savvy but he does have a newer mobile phone. He wants to be able to let other guides and operators know the status of his vessel. Especially if he is inbound or outbound.

Marie is 38 and maintains several different guiding jobs throughout the year. She is very technically savvy. She prefers larger displays and so tethers wifi to a laptop that is hard mounted in the wheelhouse of the boat she operates.

Both guides are typically quite busy tending to guests and so prefer performing actions as quick and efficiently as possible.

User Stories

1) As Bob, I want to view a list of boat statuses so that I know at a glance what status each boat is in.

Acceptance Criteria [Examples]
Given that I am Bob

And I need to review the statuses of all the boats
When I go to the FishFry Tours website
I can see all the boats in their respective swim lanes

2) As a FishFry Tours Operator, I would like to be able to create new cards for boats to describe what status they are in and be able to move them between different statuses/swim lanes. (Create/Update)

Acceptance Criteria [Examples]
Given that I am an operator
And I need to create a new card for a boat
When I select the 'Add Boat' button
I am able to add a boat through a form and it is placed into the leftmost swimlane by default

Assessment Scoring

General assessment
We will be assessing additional factors not listed such as general handling of the Story Card.

Back-end component assessment

Rating	**Looking For**
Good/Acceptable	- Loads without errors - has multiple modules/components/classes - implement error handling - RESTful - Good formatting and comments - Can list, view, create, update, delete items
Weak/Poor (Fail)	- Copied solution or tutorial with little or no changes - Does not meet the requirements for 'Good'

Front-end component assessment

Rating	**Looking For**
Good/Acceptable	- Loads without errors - has multiple modules/components/class - implements data binding - Mobile friendly (e.g. use of a CSS framework) - Good formatting and comments - Can list, view, create, update, delete items

Weak/Poor (Fail)	- Copied solution or tutorial with little or no changes - Inline CSS and JS - Does not meet the requirements for 'Good'

QA test assessment

Rating	**Looking For**
Good/Acceptable	- Executable test plans & scripts that will ensure applications meet business requirements, system goals, and fulfill end-user requirement. - Sufficient amount (>5 **valuable** unit tests) of test coverage (TDD) using a modern testing framework. - Written paragraph on what approach you would take for writing functional tests for this application
Weak/Poor (Fail)	- Tests are not executable or do not pass. - Does not meet the requirements for 'Good'

DevOps Pipeline Assessment

Rating	**Looking For**
Good/Acceptable	- The pipeline checks out code from a public repository - The pipeline checks builds code (if applicable) - The pipeline sets configuration settings - The pipeline deploys the app to a cloud-hosted environment
Weak/Poor (Fail)	- Automated pipeline does not work or contains manual steps - Relies entirely on bespoke scripts - Does not meet the requirements for 'Good'