IS27 Full Stack Developer - Technical Assignment

Requirement:

Build a simple web application that tracks and manages delivery trucks 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)
- Database Development

* ***you MUST have a backend component completed and functional. A front end component is highly recommended.***

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
- All links must be available for 10 business days after the day of your submission

Back-end Component

Build a Web service using a server-side framework (e.g. .NET, Java, Python etc) and host it in a location that is accessible from the internet.

User Authentication/Authorization is **not** required for this code challenge.
Include a list of the backend API endpoints in the README file.

Front-end Assessment

Build a Web App **using a JavaScript framework** (e.g. REACT, Angular, Vue etc) and host it in a location that is accessible from the internet.

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 delivery truck tracker.

Database Assessment

Use a relational database (e.g. Postgres, Oracle etc) to keep track of the state of application. Please provide all SQL seed files.

Providing an ER diagram is recommended.

Background:

BCM Deliveries is a small delivery startup based out of Victoria, British Columbia. With the increased demand of last mile delivery services from an uptick in the popularity of online shopping, they are looking to assist a larger delivery company and are based out of a shared warehouse. In total, they have 8 delivery trucks and 12 drivers. At any given time there are at least 4 or 5 trucks out doing deliveries.

The control office maintains a kanban-like control chart on a white board which describes the state of each of the delivery vehicles. Some of the swimlanes are ‘Loading at warehouse, Outbound for deliveries, Returning to warehouse, and Maintenance.

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

One of the drivers is a web developer during the weekends and offered to build the app. They insist on building it using an Agile Approach.

The other drivers have varying computer skills. They mainly want to see the status of all the other delivery vehicles at a glance and be able to move their cards into different swimlanes as needed.

Personas:

Bob is 26 and has been a delivery driver for most of his life. He has a dog named Wilfred that accompanies him in the delivery truck. Bob is not very technically savvy but he does have a newer mobile phone. He wants to be able to let other drivers and operators know the status of his delivery vehicle.

Marie is 38 and works at the headquarters. She is very technically savvy. She prefers larger displays and uses a large monitor to view the control chart.

The delivery drivers are typically quite busy delivering packages and so prefer performing actions as quickly and efficiently as possible.

User Stories:

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

   Acceptance Criteria [Examples]
   Given that I am Bob
   And I need to review the statuses of all the delivery trucks
   When I go to the BCM Deliveries website
   I can see all the trucks in their respective swim lanes

2. As a BCM Deliveries Operator, I would like to be able to create new cards for trucks 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 delivery truck
When I select the ‘Add Delivery Truck button
I am able to add a truck 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 and code organization.

Back-end component assessment

Rating	**Looking For**
Good/Acceptable	- Loads without errors - has multiple modules/components/classes - implement error handling - RESTful - Is good quality code (e.g. 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 - Mobile friendly (e.g. use of a CSS framework) - Is good quality code (e.g. good formatting and comments) - Can list, view, create, update, delete items - Interface works intuitively
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 fulfil end-user requirements. - 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'

Database assessment

Rating	Looking For
Good/Acceptable	- The database components are well structured and properly related. - Database tables are created and named in a logical manner.
Weak/Poor (Fail)	- A relational database is not used. - Does not meet the requirements for 'Good'