**PRACTICAL EXAMINATION 02**Subject code: FER202 – SU25   
Duration: 85 minutes

You need to create a motorbike management application using JSON Server, ReactJS, Hooks, React Router, and React Bootstrap. The application should include the following functionalities:

1. Create a ReactJS Application. (0.5 marks)

* Initialize a new ReactJS project named **fer202-02**

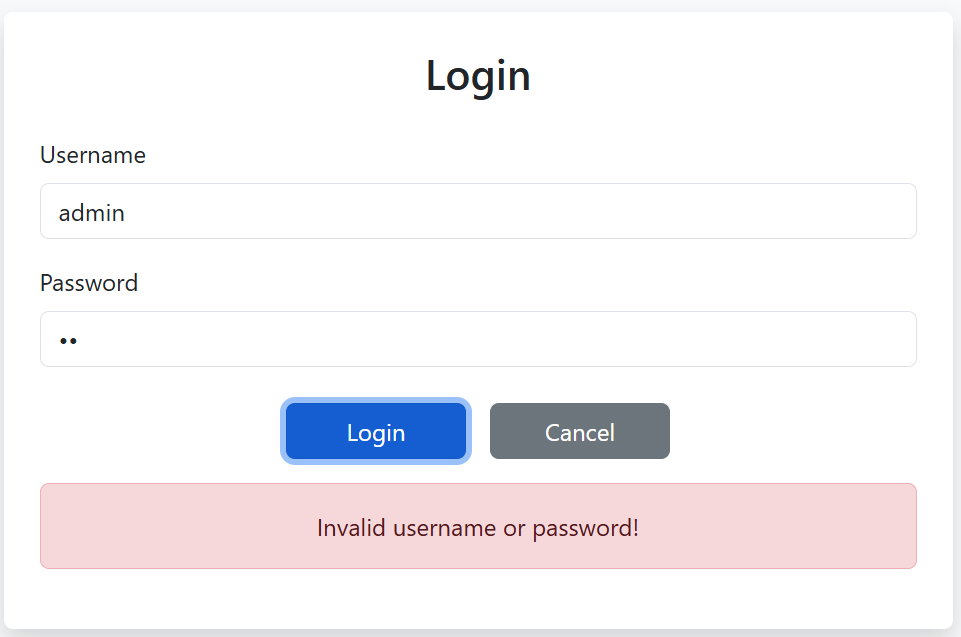
1. Create and configure JSON Server (0.5 marks)

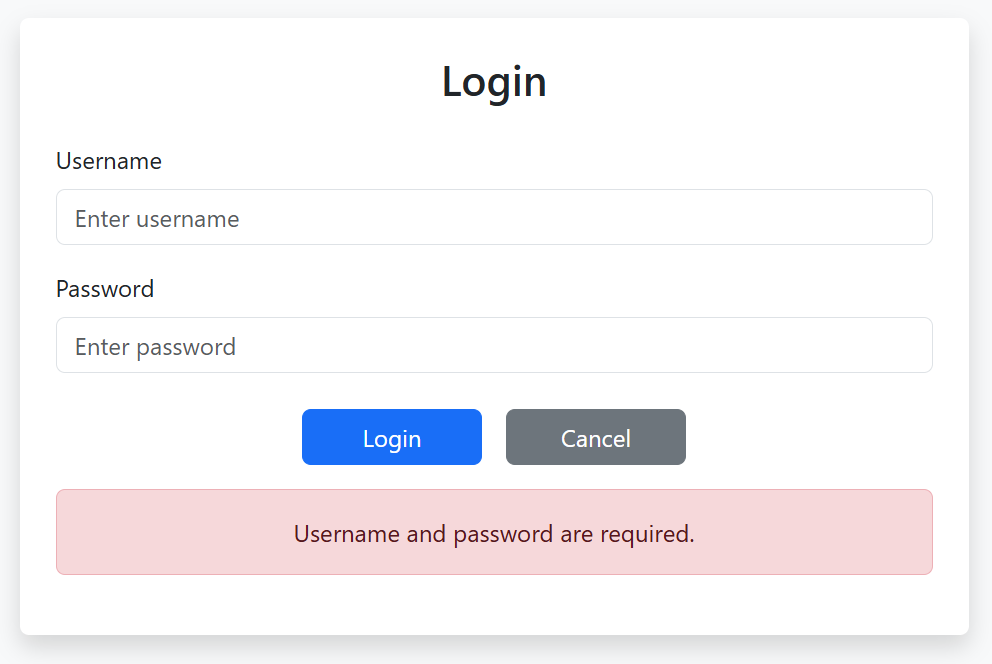
* Create a db.json file in the root folder to initialize a mock server with the motorbike and user account data as file attachment in Given folder.

1. Create a Login Form. (3 marks)

* Create a login form with 2 input fields: **Username**, **Password**. (2 marks)
  + Check the validity of the login credentials.
  + If the user enters the correct **username** and **password**, a modal should appear displaying the message 'Welcome, <username> login successful!'.

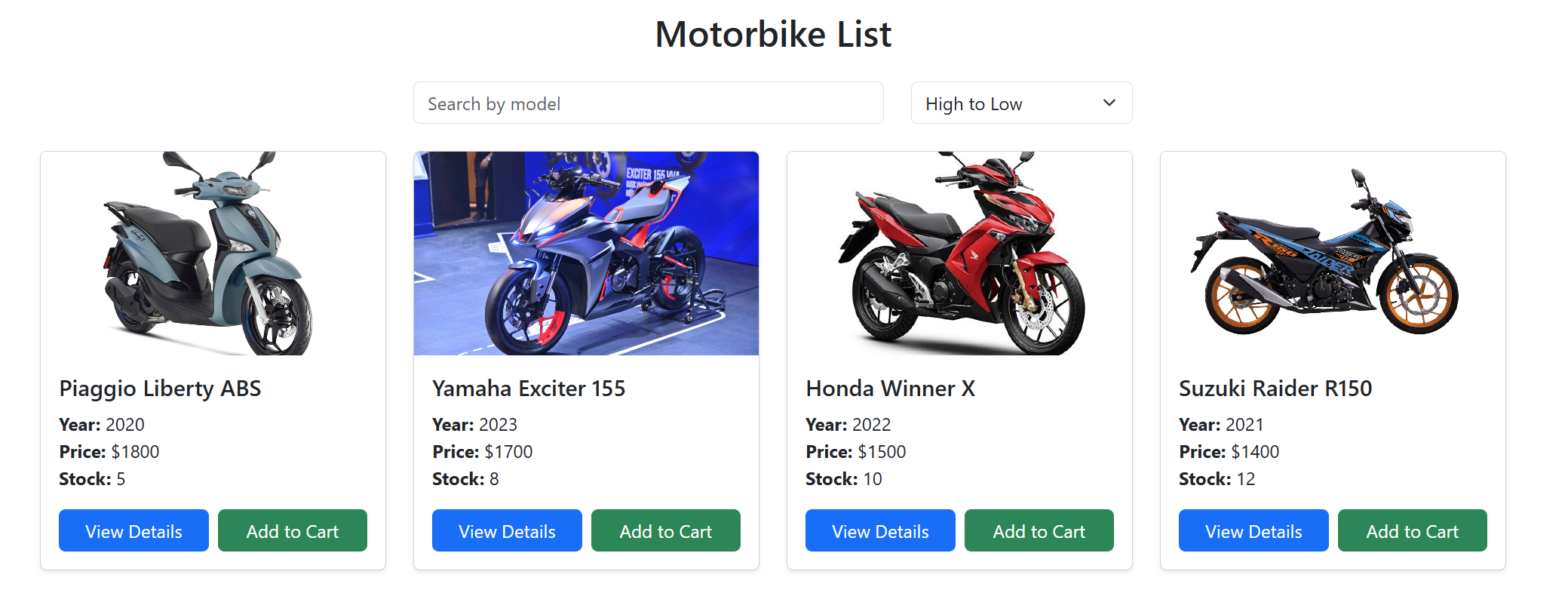
Then redirect to the motorbike list page.

* + If the username or password is incorrect, an **alert** should appear with the message **"Invalid username or password!"**.
  + Clicking the Cancel button clears all input fields and removes any displayed alerts
* Use PropTypes to ensure: (1 mark)
  + The **setUser** function is passed correctly as a required prop.
  + The username and password fields are both strings and are required.

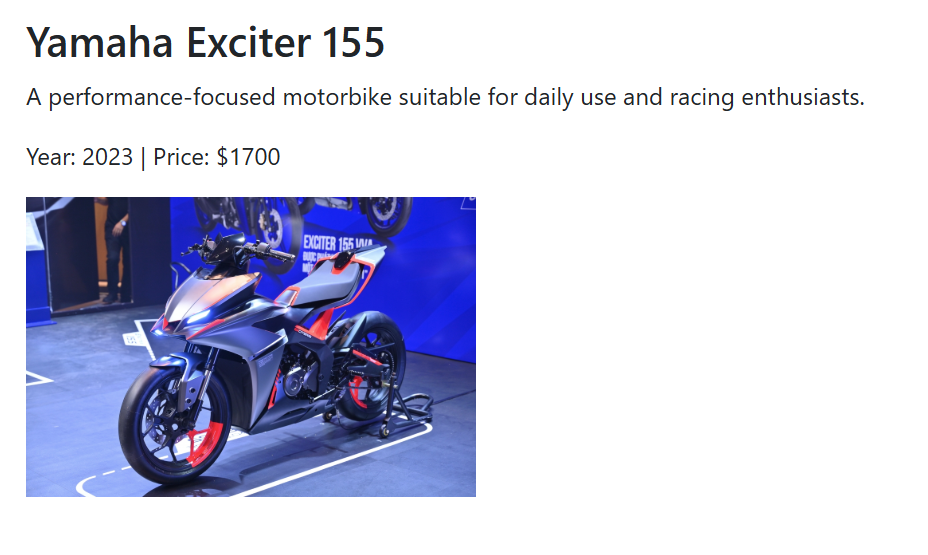


1. Create Motormotorbike Management Form (5 marks)

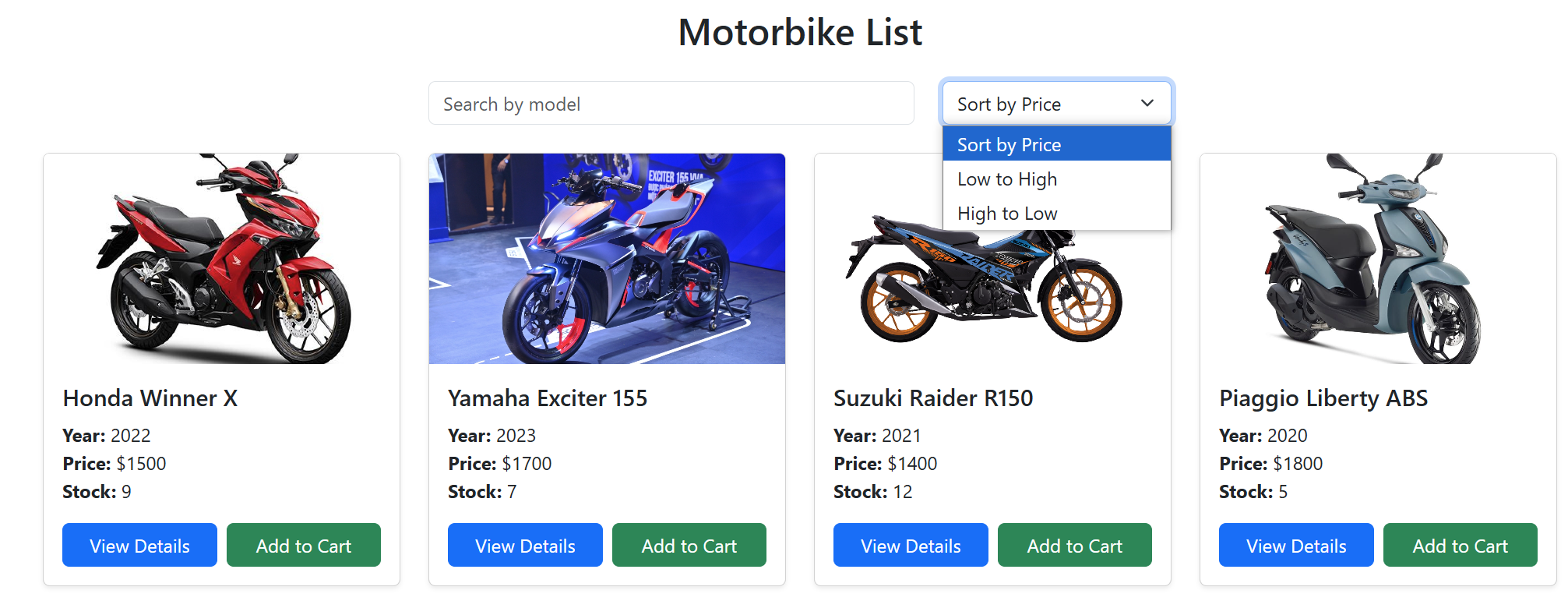
* Use React Router with routes (1 mark)
  + /motorbikes
  + /view/:id
  + /cart
* Motorbike list page (/motorbikes) (1 mark)
  + Fetch and display motorbike details (brand, model, year, price, image) from the JSON Server.



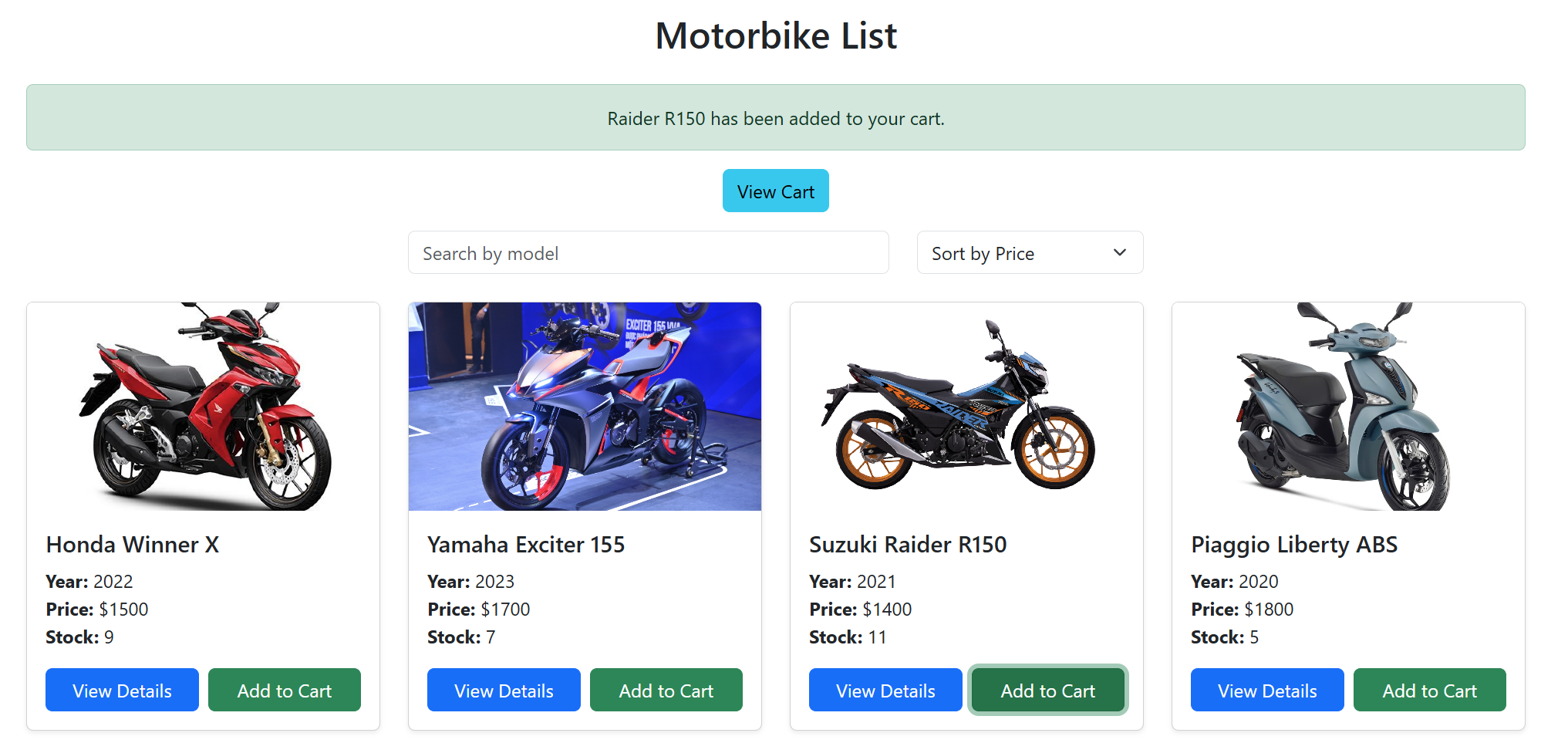
* View motorbike page (0.5 mark)
  + Each motorbike in the list should have a "View Details" button.
  + When clicked, it should display the information of the motorbike.
  + The URL should contain the motorbike id as a parameter, e.g., /motorbikes/:id.
  + If the motorbike with the given id is not found, show a 404 Not Found Page.



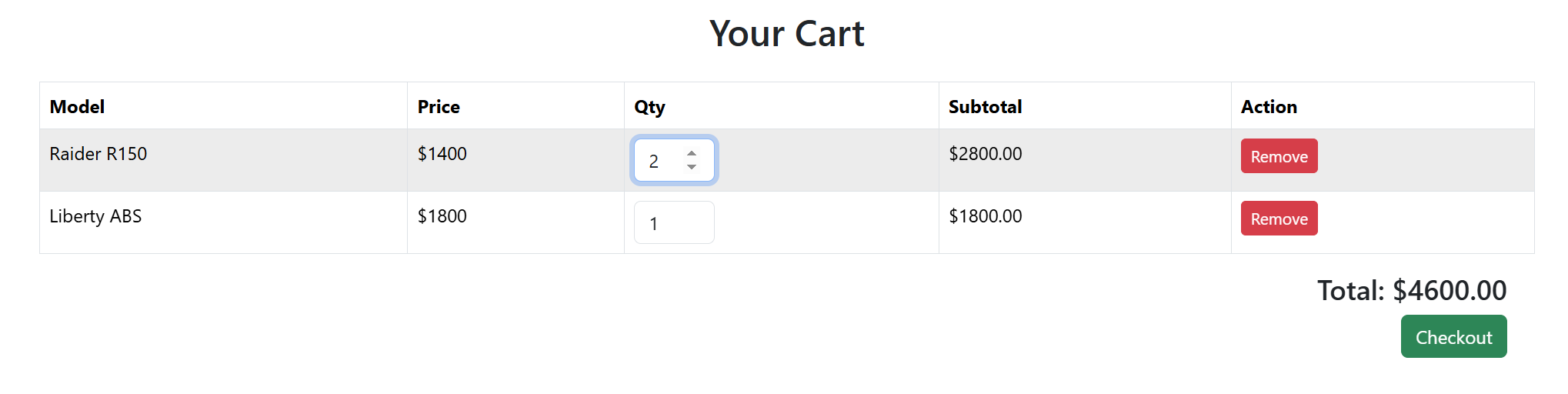
* Search by Name and Sort by Price (1 mark)
  + Search the motorbike by Model (0.5 mark)
  + Sort by price by Price (0.5 mark)



* Add to Cart (1 mark)
  + Add the selected motorbike to cart context/state.
  + Decrease motorbike stock by 1 in JSON Server.
  + Clicking the View Cart button redirects to cart page.



* Cart Page (/cart) (0.5 mark)
  + Display items in the cart: Title, quantity, subtotal, and total amount.
  + Update the quantity of the motorbike in the cart
  + Option to remove motorbike from cart (restores stock).



**Technical Requirements (1 mark)**

* Use **React Bootstrap** to style the interface and ensure that the application is responsive on mobile devices.
* Ensure that all components are properly validated using PropTypes.
* Use axios or fetch to interact with JSON Server.
* Use useReducer and useContext to manage:
  + Cart state
  + Global bike list

**Notes:**

* Students need to install and configure all required packages before the test, such as: json-server, react-router-dom, react-bootstrap bootstrap, prop-types, axios, …
* Students are allowed to use resources on their computer, but may only use the Internet to submit their exam.
* If identical test results are detected, the student's work will receive a score of zero.
* If the answer to each question is incorrect or doesn't produce the correct result, the student's work will receive a score of zero.

You must submit a folder that includes your **project folder** (**excluding the node\_modules folder**) and **a Word file named "fer202"** that contains snapshots of all your code and results related to each question. Then, compress the folder into a ZIP file named "**fptu\_fer202.zip**" and upload it to the PEA application.