

Fortnox Car Rental

Before you begin to write any code, please let us know when to expect a result. Even if this isn't a test to measure how fast you can solve the task, it's usually a good idea to have a target date to aim for.

How to get started

You may start off from a clean slate or you may choose to use the following boilerplate project for getting a UI, backend, and a database up and running quickly.

<https://github.com/noxreview/bootstrap-project>

How to send your Car Rental application

The application should be uploaded to <https://github.com/> in a **private** repository. When you are ready, share this repository with us by going to Repository Settings → Collaborators → Add People and input **noxreview** → Click “Add noxreview as a collaborator”.

When that is done, send an email to the recruiter to let them know you are done. Make sure to include your **GitHub username** in the email.

Application description and requirements

This specification describes a tiny application for renting cars. The application shall consist of two simple views, one table, and a form.

View A - Form (Rent car)

- A form containing X fields, a button, and a cost summary:
 - Which car you want to rent (dropdown / select list)
 - Which date you want to pick up the car (date picker)
 - Validation: Can't pick a date in the past
 - Which date you want to return the car (date picker)
 - Validation: Can't be a date before or on the pick-up date
 - Name of driver (text input)
 - Validation: Can't be a number
 - Age of driver
 - Validation: Must be 18 years or older.
 - Submit button to rent car and store values in the database
 - A summary of the expected cost in SEK. The cost text is not a form field, it's just information for the user placing the order.

All fields above are required to contain valid values. Suppose a user tries to submit the form

with empty input fields or invalid values. In that case, the user shall be presented with some error message indicating required fields or with a message explaining why the validation failed.

You should be able to select any of the following cars with their associated price:

- Volvo S60, 1500 kr/day
- Volkswagen Golf, 1333kr/day
- Ford Mustang, 3000kr/day
- Ford Transit, 2400kr/day

Bonus points: You can't rent a car that has already been rented out during one or more days of the period.

View B - Overview of rented cars

This view is for the car rental owner, so that he or she can easily get an overview of which cars are rented, to whom and what the profit will be from those rentals. The view should consist of a table presenting each of the following columns:

- Name of the driver
- Car
- From and to date
- Revenue

At the bottom or below the table the total revenue should be summed and presented.

When more cars are rented, the summary should automatically be updated and based on values stored in the database.

Technical requirements

The UI should be served at localhost and provide the following relative URL:s

- [/rent](#) - user ends up in the car rental form view
- [/admin](#) user ends up in the overview of rented cars.

This task should be solved using some form of MVC-style design pattern, where presentation and business logic are somewhat separated. It should be a single-page application, communicating with the backend through a REST API.

You are required to write your own SQL statements (that is, stay away from ORMs in this assignment).

The following technologies should be used for the task:

- SQL
- Java

- HTML
- CSS
- Javascript
- React

Write unit/component tests for the application. We value high coverage and clean easy-to-read tests.

If required, please supply a README.md file to describe if any manual setups are required to launch your completed project or to execute any tests.

Thanks and best of luck //Fortnox