Milestone 6

Arleth Martinez

Rest API

Full API demonstration here-
https://www.loom.com/share/52ae3e1a13b6491597e04f2a5b3568a4

Summary

- In this Milestone, I created an Express Cars API
- The API connects to MYSQL database
- Postman was used to test
- This API has the below entry points
    - GET /cars
    - GET /cars/make
    - GET /cars?carId=number
    - POST /cars
    - PUT /cars
    - DELETE /cars/carId

Challenges

One big challenge I faced in this Milestone was understanding how parameters in the URL are being passed to the controller methods. This in turn affected my postman testing. With much research + trial and error, I was finally able to successfully test.

Lessons Learned

I learned how parameters are passed from URL to methods. They are passed via query string (optional parameter) or variable segment. Express will make that variable segment available to our application with a key property that matches the variable segment name.

Ex:

Query String

localhost/cars?carId=12

Variable Segment

localhost/cars/12

Pending Bugs/ Known issues

- No pending issues at the moment

Angular Application

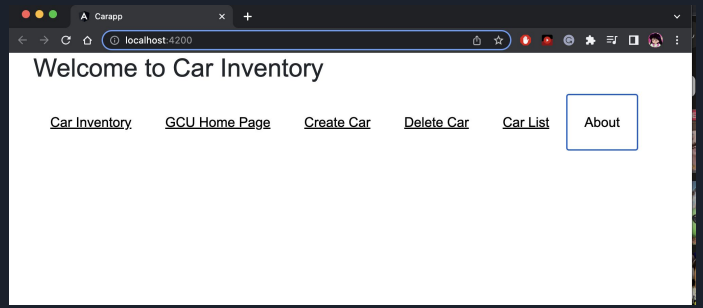
Full App demonstration here:
https://www.loom.com/share/499851ad8b774861a88ab392e97ad9a0

Summary

Express
{API}

- In this Milestone, I created an Angular Application
- The App connects to our Express API built in Milestone 3
- Via UI you can perform below actions:
  - Create acar
  - Read cars or just one car
  - Update a car
  - Delete a car

App Homepage:



Challenges

One big challenge I faced in this Milestone was creating the update capability via the UI. To do this I wanted to display the current model values rather than have the user re-input everything. I struggled with passing the selected car current values to the edit component.

Lessons Learned

I learned how to bind the model with the ngModel. This is important because any updates done in the UI, can passed back to the object.

Example:

```
<label for="Make">Make</label> <input id="Make" type="text"
class="form-control" aria-describedby="CarMake"placeholder="{{
car!.make }}" required [(ngModel)]="car!.make" name="make">
```

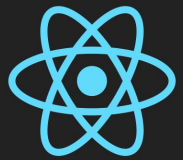
Which corresponds to:

```
@Input() car: Car | null = null;
```

Pending Bugs/ Known issues

- No pending issues at the moment

React Application



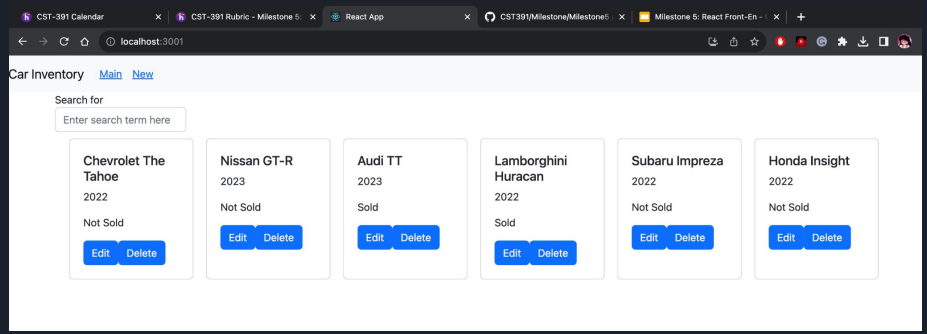
Full App Demonstration here:
https://www.loom.com/share/432d61155bdb44c796f6b45465ce5b2b

Summary

Express
{API}

- In this Milestone, I created a React Application
- The App connects to our Express API built in Milestone 3
- Via UI you can perform below actions:
  - Create a car
  - Read cars
  - Update a car
  - Delete a car

App Homepage:



Challenges

One big challenge I faced in this Milestone was creating the delete capability via the UI. To do this I had to pass the car ID to the parent delete function and call a delete request via Axios.

Lessons Learned

I learned the significance of props which are used to pass data from a parent component to a child
Example:

```
function Car(props) {

  return <h2>I am a { props.model }!</h2>;

  }
```

State is similar but manages the data within the component:

Example:

```
const [carList, setCarList] = useState([]);
```

Pending Bugs/ Known issues

- No pending issues at the moment

Christian Worldview

Website Accessibility - websites and web tools are properly designed and coded, people with disabilities can use them. Many sites and tools are developed with accessibility barriers. We should strive to make our application accessible to all.

Best Practices:

- Write alt text for images.
- Use heading levels on text.
- Caption, transcribe, and describe videos.
- Label and style links clearly.
- Specify the page language in the code.