Car Dealership Review Platform

Major Project
By Abir Sait

Table of contents

01 Overview

Overview

Objective

The project aims to build and develop a cloud-native web application for users to review car dealerships.

1 Frontend: HTML, CSS, React.js

2 Backend: Python, Node.js, Express.js, Django

3 Database: MongoDB

4 Microservices: RESTful APIs

5 Cloud Deployment: IBM Code Engine, Kubernetes

02 Introduction

Introduction

- This project is part of the Coursera-guided "Full Stack Application Development Capstone Project," which provides step-by-step instructions, code snippets, and hints.
- The course focuses on building a comprehensive full-stack web application using Django, React, Express, and MongoDB.
- The Car Dealership Review Application allows users to view and submit reviews, helping dealerships understand customer feedback and improve their services.

03 Course Summary

Course Summary

- This project is based on the Coursera-guided "Full Stack Application Development Capstone Project," which provides step-by-step instructions, code snippets, and hints.

- This course helped me:
  - Learn to build comprehensive full-stack web applications using Django, React, Express, and MongoDB.
  - Develop skills in creating dynamic front-end pages, managing users, and implementing API endpoints.
  - Gain experience in deploying applications using Docker, Kubernetes, and IBM Cloud.
  - Implement CI/CD pipelines to automate testing, linting, and deployment processes.

04 Project Summary

Project Summary

Objective:

Develop a cloud-native, full-stack application that allows users to review and rate car dealerships. The app integrates multiple technologies and frameworks to provide a seamless user experience.

Module 1 Updated static web pages with Django and Bootstrap

Module 2 Managed user authentication with React and Django, resolving browser compatibility issues.

Module 3 Created API endpoints with Express and MongoDB, and deployed a sentiment analysis service on IBM Cloud.

Module 4 Added dynamic front-end pages with React, integrating them with backend services.

Module 5 Implemented CI/CD with GitHub Actions, containerized the application with Docker, and deployed it using Kubernetes

05 Problem Statement

Problem statement

Challenge: Develop a full-stack web application that enables customers to:

- View Dealership Information:
  - Access a list of dealerships.
  - Filter by state.
  - View detailed reviews.
- Submit Reviews:
  - Create accounts.
  - Submit reviews with detailed attributes.
  - View submitted reviews.
- Manage Dealership Data:
  - Administrators manage dealership attributes.
  - Include car makes and models.

Key Steps:

- Create and style static web pages using Bootstrap.
- Implement user management with Django's authentication and a React front-end.
- Develop backend services with Node.js and Express to manage dealer and review data.
- Containerize services using Docker.
- Deploy a sentiment analysis service on IBM Cloud Code Engine.
- Set up CI/CD pipelines with GitHub Actions.
- Deploy the application using Kubernetes.
- Thoroughly test each component to ensure functionality and reliability.

06 Design

Design Architecture

Frontend

React.js for building dynamic user interfaces.

Backend

Django for handling user management and static content.
Express.js and Node.js for developing RESTful API services.

Database

MongoDB for storing dealership and review data.

Deployment

Docker for containerization.
Kubernetes for orchestration.
IBM Cloud for hosting and scaling services.

Design Architecture

Components

Static Pages: Created with Django, styled using Bootstrap.
User Management: Handled by Django, interfaced with React frontend.
API Services: Developed with Express, integrated with MongoDB.
Sentiment Analysis: Deployed on IBM Cloud Code Engine for analyzing review sentiments.

Microservices Approach:

Independent components for API, user management, and sentiment analysis.Each service is containerized and deployed independently, allowing scalability and maintainability.

07 Implementation

With tools used

Module 1: Static Pages

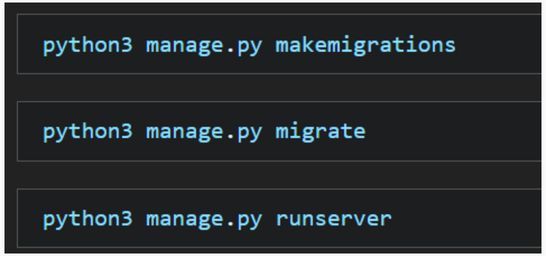
To handle user management, car models, and routes MongoDB services for dealership and customer reviews.

Tasks:

- Fork and clone the project repository from GitHub.
- Run the Django app on the development server.
- Add Bootstrap navigation to the website.
- Create "About Us" and "Contact Us" static pages.

Module 1: Static Pages

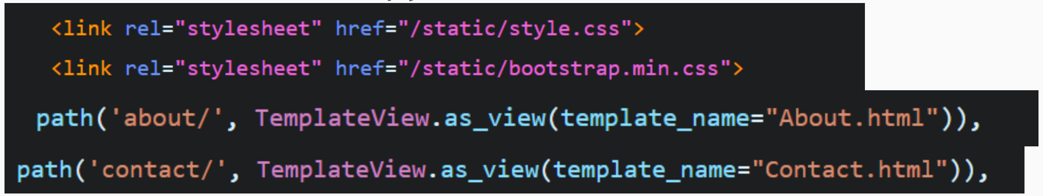
To handle user management, car models, and routes MongoDB services for dealership and customer reviews.



```
python3 manage.py makemigrations

python3 manage.py migrate

python3 manage.py runserver
```

Steps:

- **Fork & Clone:** Create a personal copy and clone it locally.
- **Run Django App:** Set up and start the development server.
- **Bootstrap Navigation:** Integrate and update templates with navigation.
- **Static Pages:** Develop HTML templates for "About Us" and "Contact Us" pages and route on urls.py



```
  <link rel="stylesheet" href="/static/style.css">
  <link rel="stylesheet" href="/static/bootstrap.min.css">
 path('about/', TemplateView.as_view(template_name="About.html")),
path('contact/', TemplateView.as_view(template_name="Contact.html")),
```

Module 2: User Management

Implement user roles by adding authentication and authorization features to manage access

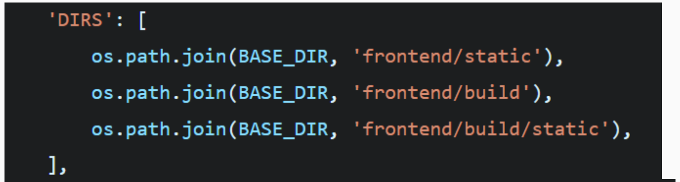
Tasks:

- Create a superuser for your app.
- Build the client side and configure it.
- Check the client configuration.
- Add a login view to handle login requests.
- Add a logout view to handle logout requests.
- Add a registration view to handle sign-up requests.

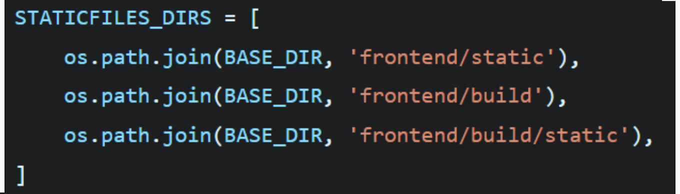
Module 2: User Management

Steps

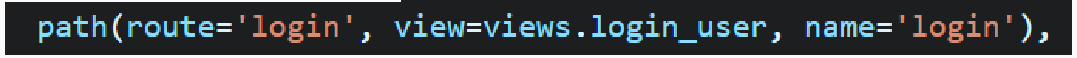
- Use python manage.py createsuperuser to establish admin access.
- Set up the frontend (e.g., React) and ensure API communication with Django.
- Test the frontend and ensure seamless interaction with the backend.
- Implement login functionality using Django's authentication.
- Handle logout requests and redirect users to the login page.
- Create a sign-up form and handle new user registrations.



```
    'DIRS': [
        os.path.join(BASE_DIR, 'frontend/static'),
        os.path.join(BASE_DIR, 'frontend/build'),
        os.path.join(BASE_DIR, 'frontend/build/static'),
    ],
```



```
STATICFILES_DIRS = [
    os.path.join(BASE_DIR, 'frontend/static'),
    os.path.join(BASE_DIR, 'frontend/build'),
    os.path.join(BASE_DIR, 'frontend/build/static'),
]
```

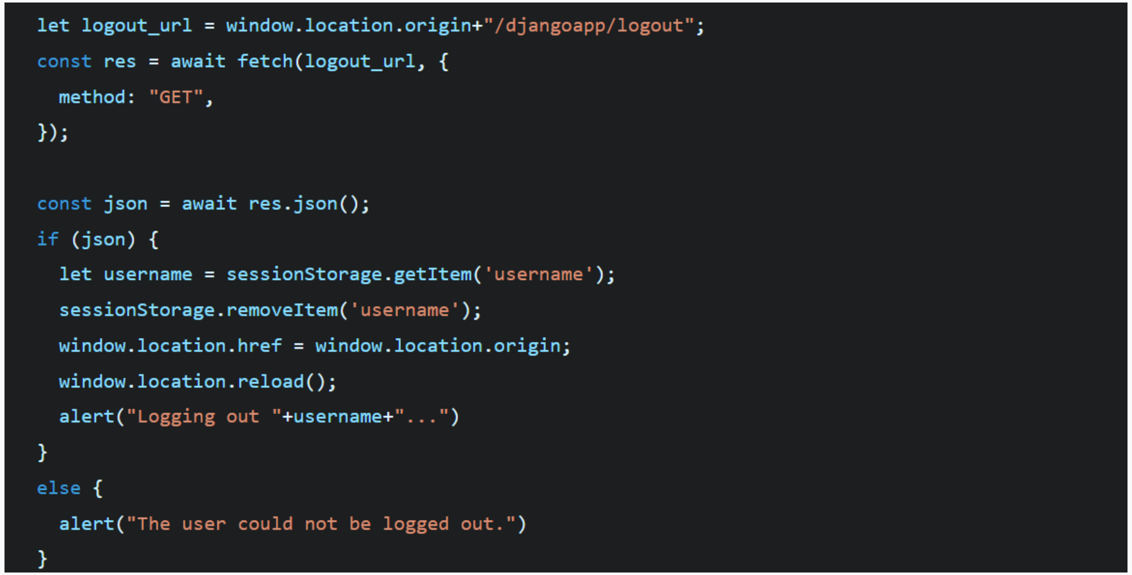


```
path(route='login', view=views.login_user, name='login'),
```

Module 2: User Management

Steps

Logout functionality code snippet



```
let logout_url = window.location.origin+"/djangoapp/logout";
const res = await fetch(logout_url, {
  method: "GET",
});

const json = await res.json();
if (json) {
  let username = sessionStorage.getItem('username');
  sessionStorage.removeItem('username');
  window.location.href = window.location.origin;
  window.location.reload();
  alert("Logging out "+username+"...")
}
else {
  alert("The user could not be logged out.")
}
```

Module 3: Backend Services

Implement user roles by adding authentication and authorization features to manage access

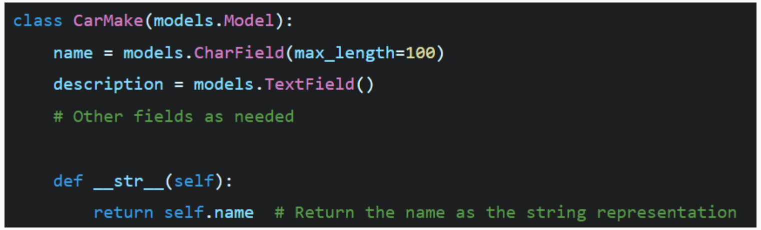
Tasks:

- Develop backend services for your Django application using JavaScript.
- Set-up a new Express MongoDB server in a Docker container.
- Deploy sentiment analyzer on IBM Code Engine.
- Create data models in your Django application.
- Create proxy services to call cloud functions in Django.
- Create CarModel and CarMake Django models
- Register CarModel and CarMake models with the admin site
- Create new car models objects with associated car makes and dealerships

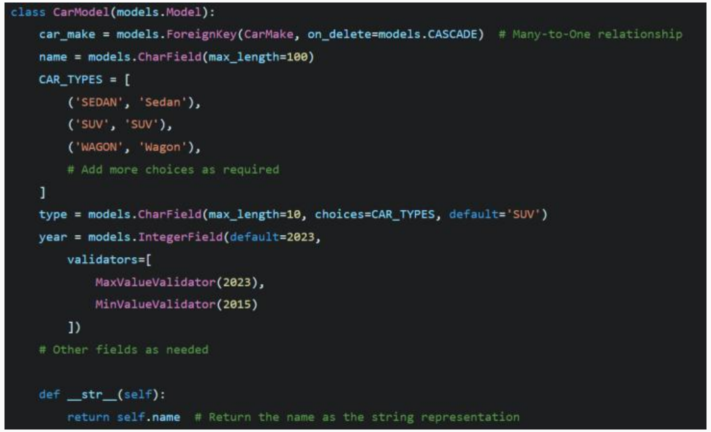
Module 3: Backend Services

Steps

- Develop Backend Services: Implement API endpoints in Express with MongoDB using Mongoose (app.js).
- Containerize and Deploy: Build and run Docker containers for MongoDB and Express (docker-compose up), and deploy the sentiment analyzer on IBM Code Engine.
- Create Django Models: Define and register CarMake and CarModel in Django, and populate the database with car data.
- Create Proxy Services: Develop Django proxy services (restapis.py), and update views (views.py) and URLs (urls.py) for dealership and review data.

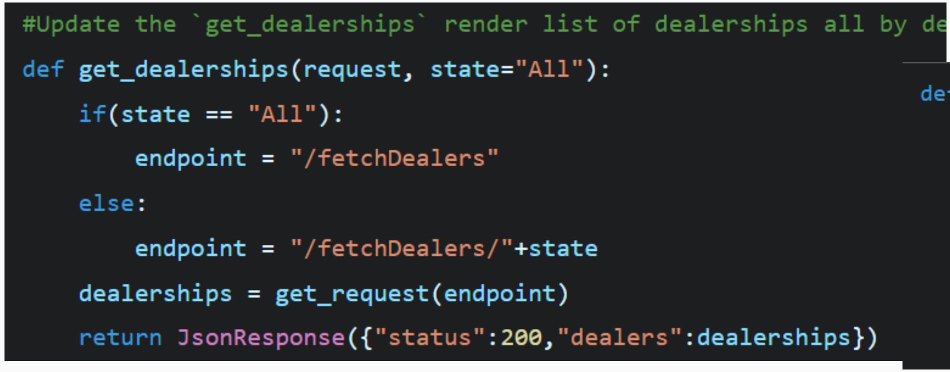


```python
class CarMake(models.Model):
    name = models.CharField(max_length=100)
    description = models.TextField()
    # Other fields as needed

    def __str__(self):
        return self.name  # Return the name as the string representation
```

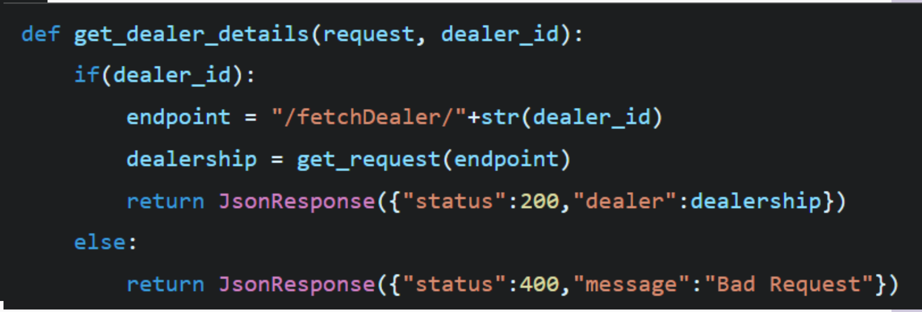


```python
class CarModel(models.Model):
    car_make = models.ForeignKey(CarMake, on_delete=models.CASCADE)  # Many-to-One relationship
    name = models.CharField(max_length=100)
    CAR_TYPES = [
        ('SEDAN', 'Sedan'),
        ('SUV', 'SUV'),
        ('WAGON', 'Wagon'),
        # Add more choices as required
    ]
    type = models.CharField(max_length=10, choices=CAR_TYPES, default='SUV')
    year = models.IntegerField(default=2023,
        validators=[
            MaxValueValidator(2023),
            MinValueValidator(2015)
        ])
    # Other fields as needed

    def __str__(self):
        return self.name  # Return the name as the string representation
```

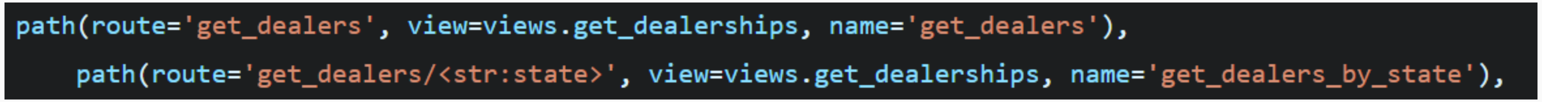
Module 3: Backend Services



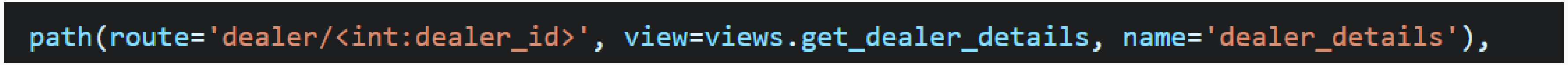
```python
#Update the `get_dealerships` render list of dealerships all by de
def get_dealerships(request, state="All"):
    if(state == "All"):
        endpoint = "/fetchDealers"
    else:
        endpoint = "/fetchDealers/"+state
    dealerships = get_request(endpoint)
    return JsonResponse({"status":200,"dealers":dealerships})
```



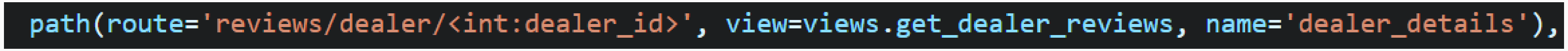
```python
def get_dealer_details(request, dealer_id):
    if(dealer_id):
        endpoint = "/fetchDealer/"+str(dealer_id)
        dealership = get_request(endpoint)
        return JsonResponse({"status":200,"dealer":dealership})
    else:
        return JsonResponse({"status":400,"message":"Bad Request"})
```



```python
path(route='get_dealers', view=views.get_dealerships, name='get_dealers'),
    path(route='get_dealers/<str:state>', view=views.get_dealerships, name='get_dealers_by_state'),
```



```python
 path(route='dealer/<int:dealer_id>', view=views.get_dealer_details, name='dealer_details'),
```



```python
 path(route='reviews/dealer/<int:dealer_id>', view=views.get_dealer_reviews, name='dealer_details'),
```

Module 4: Dynamic pages

Functionality to manage car dealerships is added including creating, updating, and deleting dealerships, as well as listing them for users to view.

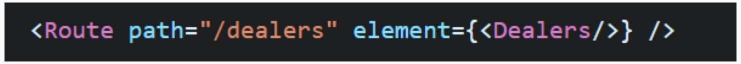
Tasks:

- Build frontend pages to present backend services to end users.
- Create a component to list the dealerships.
- Develop a dealer details and reviews component.
- Create a review submission page.

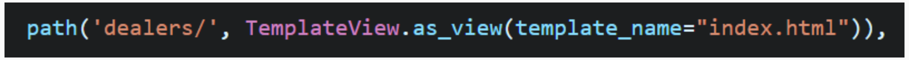
Module 4: Dynamic pages

Steps

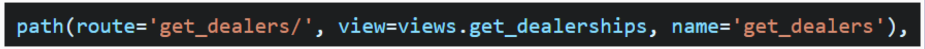
- **Set Up React Components:** Imported and added Dealers and Dealer components in App.js.
- Configured routes in urls.py for dealers/ and dealer/:id.
- **Create Dealer Reviews Page:** Added PostReview component and configured the route in App.js and urls.py.
- Built the front end and ensured proper routing for review submissions.
- **Test Functionality:** Tested the application by logging in, submitting a review, and capturing a screenshot of the submission for verification.



```
<Route path="/dealers" element={<Dealers/>} />
```



```
path('dealers/', TemplateView.as_view(template_name="index.html")),
```



```
path(route='get_dealers/', view=views.get_dealerships, name='get_dealers'),
```

Module 5: CI/CD, Containerize & Deploy to Kubernetes

Continuous Integration (CI) and Continuous Delivery (CD) are set up for the source code. Additionally, the application is containerized and deployed to Kubernetes for better scalability and management

Tasks:

- Enable GitHub Actions and run the Linting workflow
- Add the ability to your application to run in a container
- Add deployment artifacts for your application so it can be managed by Kubernetes

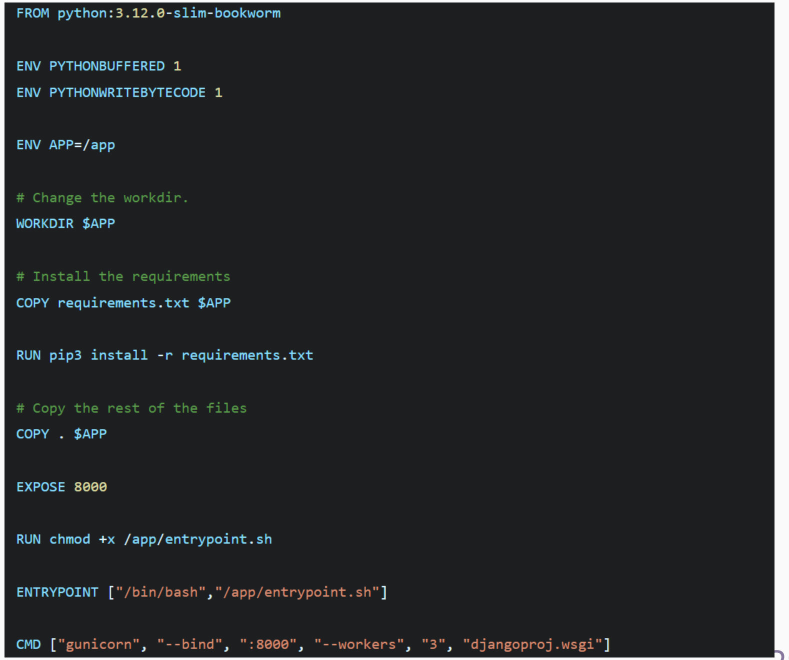
Module 5: CI/CD, Containerize & Deploy to Kubernetes

Steps

- **Dockerfile**: Created Dockerfile to define the image and an entrypoint.sh script to manage database migrations and static files.
- **Image Build & Push**: Built the Docker image and pushed it to IBM Cloud Image Registry (ICR).
- **Kubernetes Deployment:** Deployed using deployment.yaml and accessed via port-forwarding.
- **Verification**: Tested the deployed app, captured screenshots for submission.

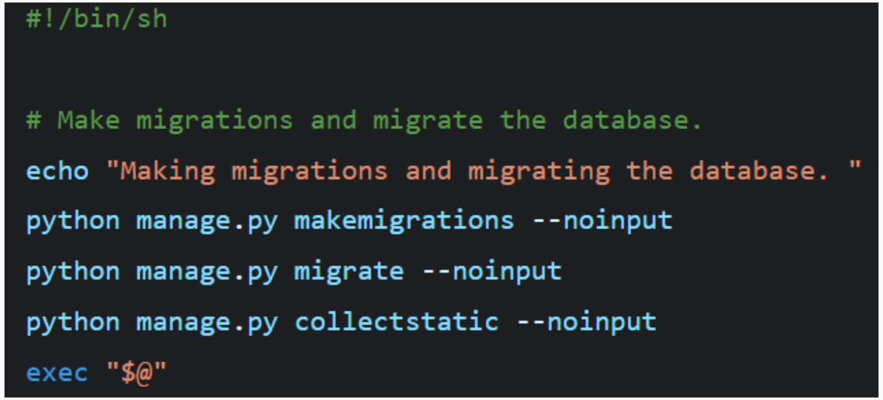
Module 5: CI/CD, Containerize & Deploy to Kubernetes

Dockerfile



```
FROM python:3.12.0-slim-bookworm

ENV PYTHONBUFFERED 1
ENV PYTHONWRITEBYTECODE 1

ENV APP=/app

# Change the workdir.
WORKDIR $APP

# Install the requirements
COPY requirements.txt $APP

RUN pip3 install -r requirements.txt

# Copy the rest of the files
COPY . $APP

EXPOSE 8000

RUN chmod +x /app/entrypoint.sh

ENTRYPOINT ["/bin/bash","/app/entrypoint.sh"]

CMD ["gunicorn", "--bind", ":8000", "--workers", "3", "djangoproj.wsgi"]
```

Entrypoint.sh



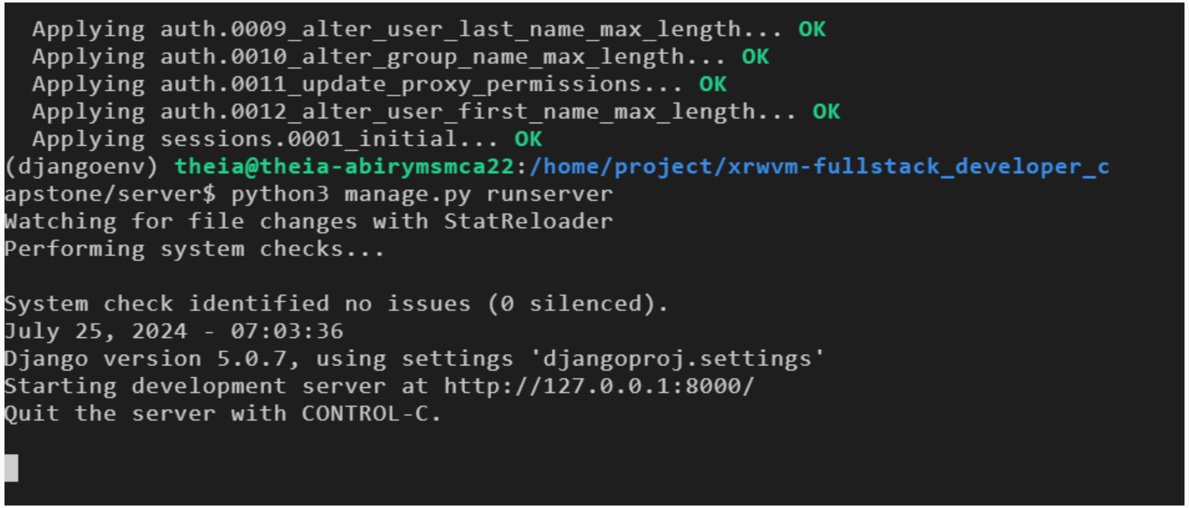
```
#!/bin/sh


# Make migrations and migrate the database.
echo "Making migrations and migrating the database. "
python manage.py makemigrations --noinput
python manage.py migrate --noinput
python manage.py collectstatic --noinput
exec "$@"
```

08 Results

Module 1: Django server

Start the Django server with **python manage.py runserver.**

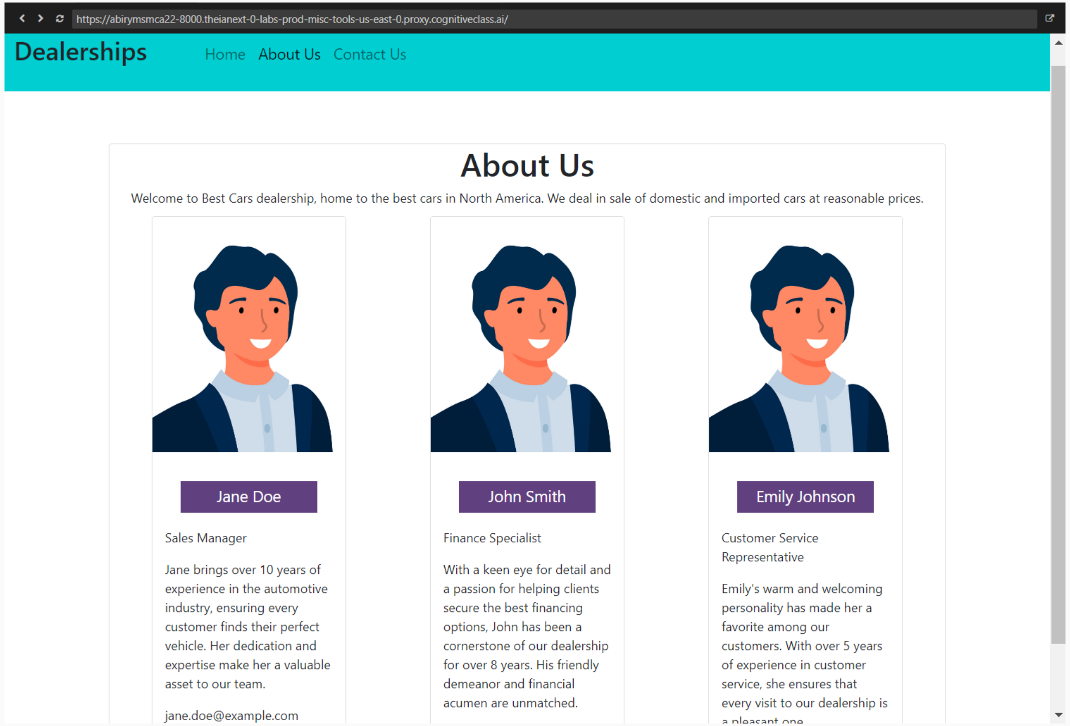
Verify that the app is running and accessible.



```
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying auth.0010_alter_group_name_max_length... OK
  Applying auth.0011_update_proxy_permissions... OK
  Applying auth.0012_alter_user_first_name_max_length... OK
  Applying sessions.0001_initial... OK
(djangoenv) theia@theia-abirymsmca22:/home/project/xrwvm-fullstack_developer_c
apstone/server$ python3 manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
July 25, 2024 - 07:03:36
Django version 5.0.7, using settings 'djangoproj.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```

Module 1: About us

The "About Us" page showcases the company's mission, values, and team information. This static component of the Django application is styled using Bootstrap, ensuring a consistent and professional appearance.


https://abirymsmca22-8000.theianext-0-labs-prod-misc-tools-us-east-0.proxy.cognitiveclass.ai/

Dealerships Home About Us Contact Us

About Us

Welcome to Best Cars dealership, home to the best cars in North America. We deal in sale of domestic and imported cars at reasonable prices.

Jane Doe

Sales Manager

Jane brings over 10 years of experience in the automotive industry, ensuring every customer finds their perfect vehicle. Her dedication and expertise make her a valuable asset to our team.

jane.doe@example.com

John Smith

Finance Specialist

With a keen eye for detail and a passion for helping clients secure the best financing options, John has been a cornerstone of our dealership for over 8 years. His friendly demeanor and financial acumen are unmatched.

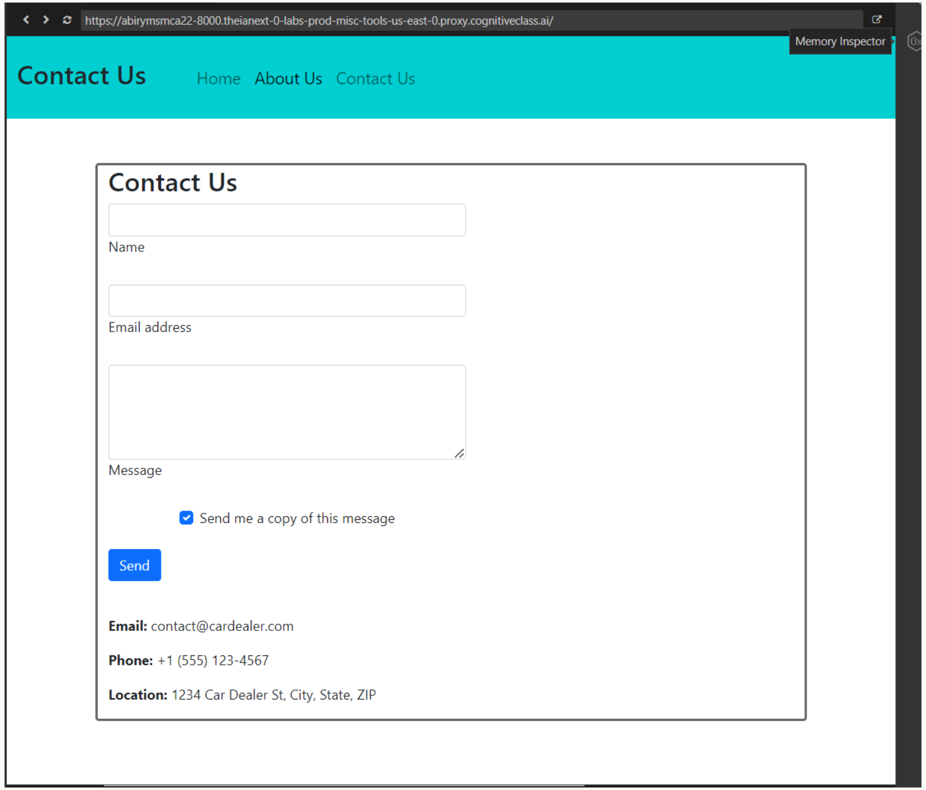
Emily Johnson

Customer Service Representative

Emily's warm and welcoming personality has made her a favorite among our customers. With over 5 years of experience in customer service, she ensures that every visit to our dealership is a pleasant one

Module 1: Contact us

The "Contact Us" page provides users with a form to reach out to the company. This feature is crucial for customer support and engagement, allowing users to submit inquiries directly through the application.


https://abirymsmca22-8000.theianext-0-labs-prod-misc-tools-us-east-0.proxy.cognitiveclass.ai/
Memory Inspector
Contact Us
Home
About Us
Contact Us

Contact Us

Name

Email address

Message

[✓] Send me a copy of this message

Send

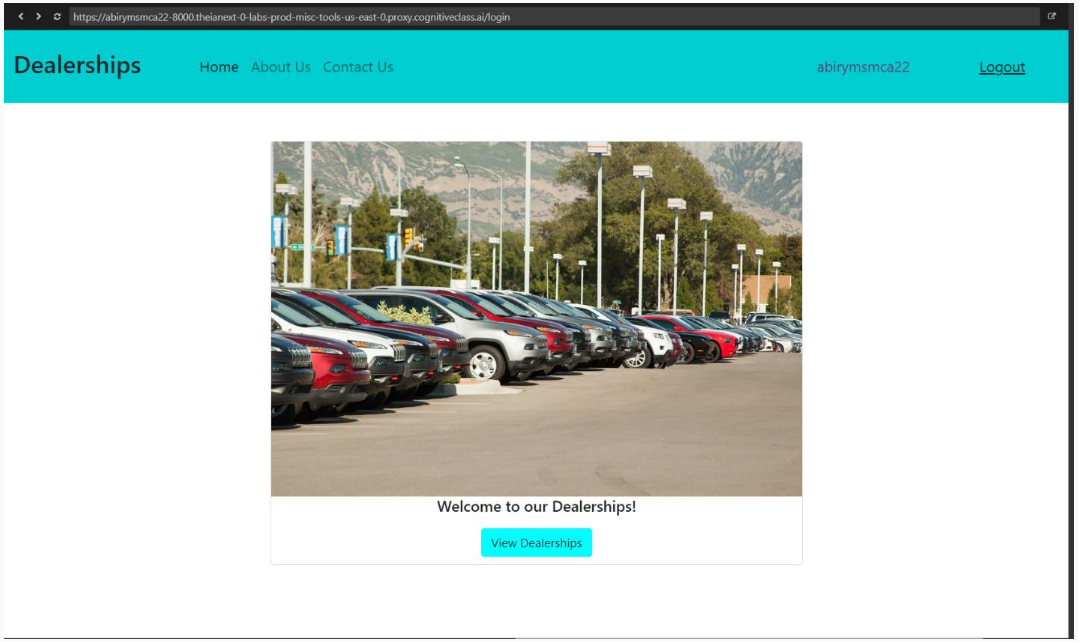
Email: contact@cardealer.com

Phone: +1 (555) 123-4567

Location: 1234 Car Dealer St, City, State, ZIP

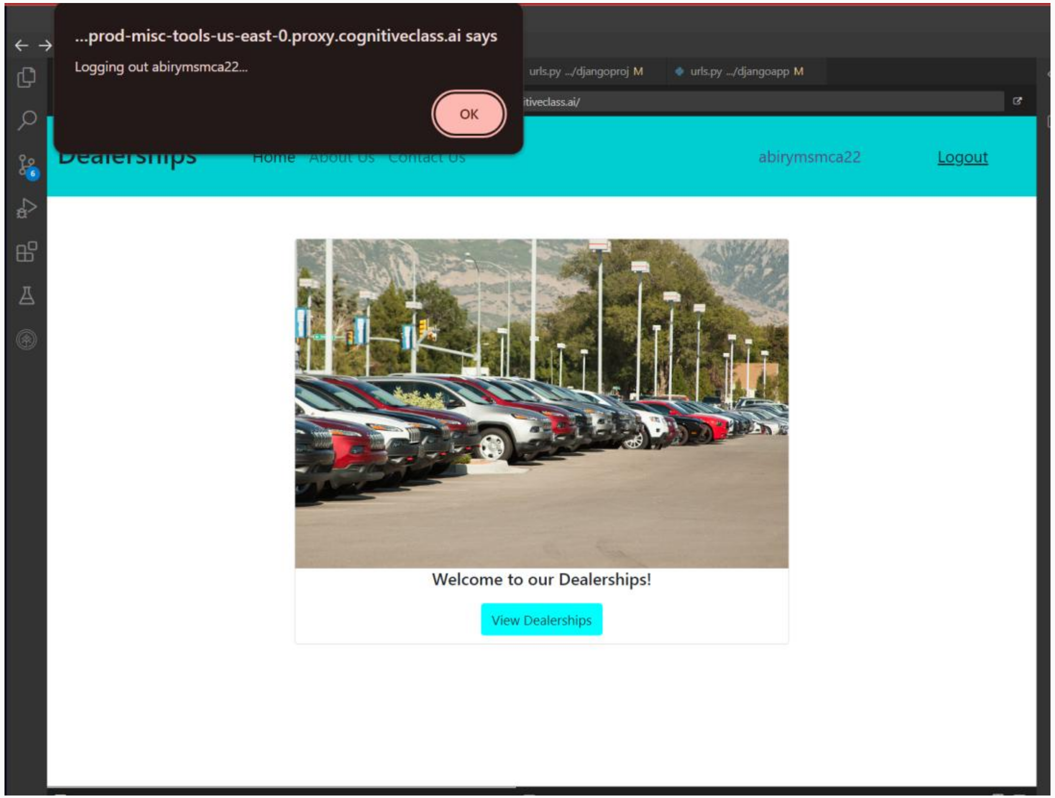
Module 2: Login

The login page allows users to authenticate themselves before accessing restricted features of the application. This task verifies the implementation of the user authentication mechanism, ensuring secure access to user-specific functionalities.


https://abirymsmca22-8000.theianext-0-labs-prod-misc-tools-us-east-0.proxy.cognitiveclass.ai/login
Dealerships
Home
About Us
Contact Us
abirymsmca22
Logout
Welcome to our Dealerships!
View Dealerships

Module 2: Logout

The logout alert confirms that the user has successfully logged out of the application, ensuring proper session management and security.

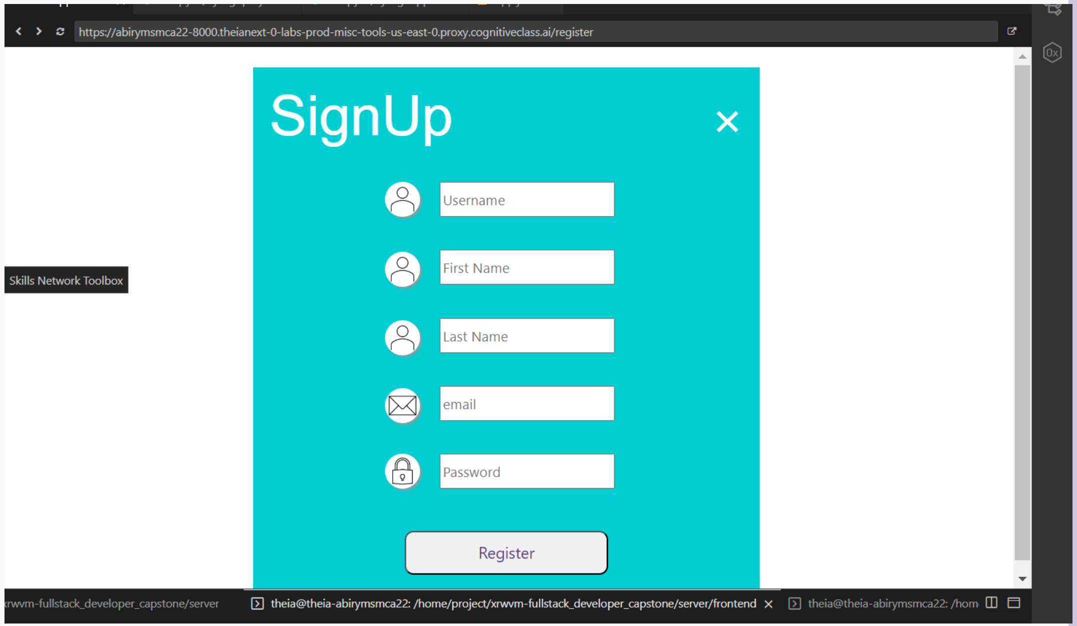


...prod-misc-tools-us-east-0.proxy.cognitiveclass.ai says

Logging out abirymsmca22...

Module 2: Register/signup

The sign-up page enables new users to create accounts. This task ensures that the user registration process is working as intended, allowing the application to expand its user base.



https://abirymsmca22-8000.theianext-0-labs-prod-misc-tools-us-east-0.proxy.cognitiveclass.ai/register

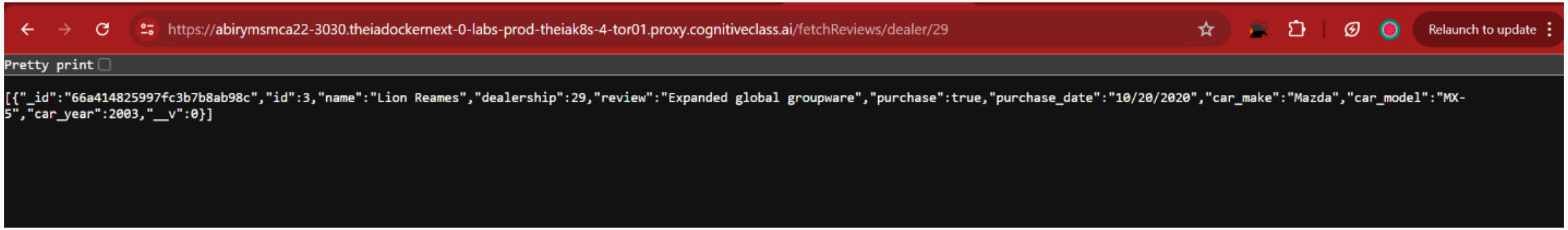
xrwvm-fullstack_developer_capstone/server

theia@theia-abirymsmca22: /home/project/xrwvm-fullstack_developer_capstone/server/frontend

theia@theia-abirymsmca22: /hom

Module 3: Dealer review

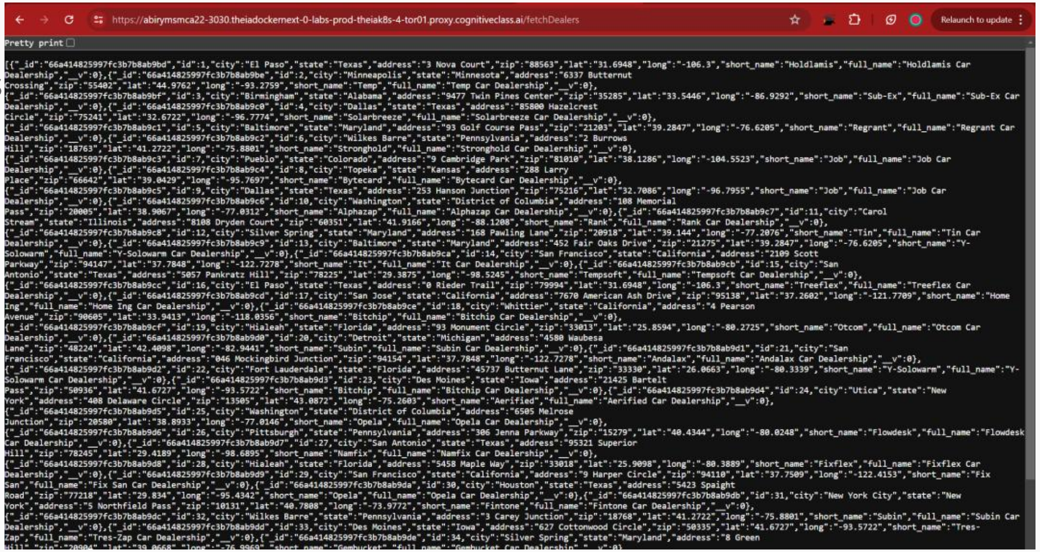
This screenshot displays the dealer reviews fetched through the Express-Mongo backend. It confirms the successful integration of the backend service with the front-end React application, demonstrating the application’s capability to handle and display user-generated content.


https://abirymsmca22-3030.theiadockernext-0-labs-prod-theiak8s-4-tor01.proxy.cognitiveclass.ai/fetchReviews/dealer/29
Relaunch to update
Pretty print

```
[{"_id":"66a414825997fc3b7b8ab98c","id":3,"name":"Lion Reames","dealership":29,"review":"Expanded global groupware","purchase":true,"purchase_date":"10/20/2020","car_make":"Mazda","car_model":"MX-
5","car_year":2003,"__v":0}]
```

Module 3: Dealer review

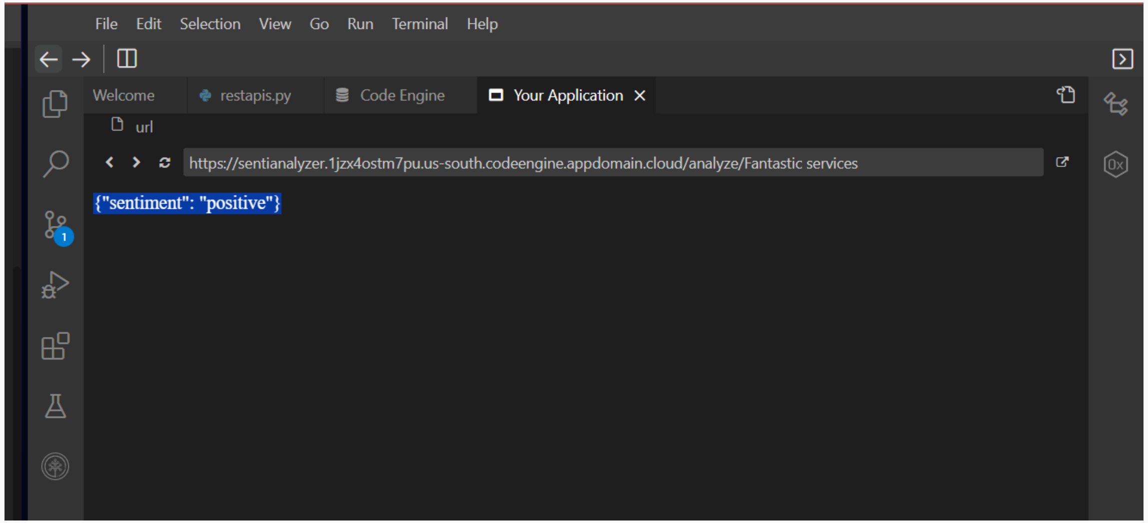
This screenshot displays the dealer reviews fetched through the Express-Mongo backend. ItThis screenshot shows a list of all dealers retrieved from the MongoDB database via the Express application. It demonstrates the correct data retrieval and display functionality, ensuring users can view comprehensive dealership information confirms the successful integration of the backend service with the front-end React application, demonstrating the application's capability to handle and display user-generated content.


https://abirymsmca22-3030.theiadockernext-0-labs-prod-theiak8s-4-tor01.proxy.cognitiveclass.ai/fetchDealers
Relaunch to update
Pretty print

```
[{"_id":"66a414825997fc3b7b8ab9bd","id":1,"city":"El Paso","state":"Texas","address":"3 Nova Court","zip":"88563","lat":"31.6948","long":"-106.3","short_name":"Holdlamis","full_name":"Holdlamis Car
Dealership","__v":0},{"_id":"66a414825997fc3b7b8ab9be","id":2,"city":"Minneapolis","state":"Minnesota","address":"6337 Butternut
Crossing","zip":"55402","lat":"44.9762","long":"-93.2759","short_name":"Temp","full_name":"Temp Car Dealership","__v":0},
{"_id":"66a414825997fc3b7b8ab9bf","id":3,"city":"Birmingham","state":"Alabama","address":"9477 Twin Pines Center","zip":"35285","lat":"33.5446","long":"-86.9292","short_name":"Sub-Ex","full_name":"Sub-Ex Car
Dealership","__v":0},{"_id":"66a414825997fc3b7b8ab9c0","id":4,"city":"Dallas","state":"Texas","address":"85800 Hazelcrest
Circle","zip":"75241","lat":"32.6722","long":"-96.7774","short_name":"Solarbreeze","full_name":"Solarbreeze Car Dealership","__v":0},
{"_id":"66a414825997fc3b7b8ab9c1","id":5,"city":"Baltimore","state":"Maryland","address":"93 Golf Course Pass","zip":"21203","lat":"39.2847","long":"-76.6205","short_name":"Regrant","full_name":"Regrant Car
Dealership","__v":0},{"_id":"66a414825997fc3b7b8ab9c2","id":6,"city":"Wilkes Barre","state":"Pennsylvania","address":"2 Burrows
Hill","zip":"18763","lat":"41.2722","long":"-75.8801","short_name":"Stronghold","full_name":"Stronghold Car Dealership","__v":0},
{"_id":"66a414825997fc3b7b8ab9c3","id":7,"city":"Pueblo","state":"Colorado","address":"9 Cambridge Park","zip":"81010","lat":"38.1286","long":"-104.5523","short_name":"Job","full_name":"Job Car
Dealership","__v":0},{"_id":"66a414825997fc3b7b8ab9c4","id":8,"city":"Topeka","state":"Kansas","address":"288 Larry
Place","zip":"66642","lat":"39.0429","long":"-95.7697","short_name":"Bytecard","full_name":"Bytecard Car Dealership","__v":0},
{"_id":"66a414825997fc3b7b8ab9c5","id":9,"city":"Dallas","state":"Texas","address":"253 Hanson Junction","zip":"75216","lat":"32.7086","long":"-96.7955","short_name":"Job","full_name":"Job Car
Dealership","__v":0},{"_id":"66a414825997fc3b7b8ab9c6","id":10,"city":"Washington","state":"District of Columbia","address":"108 Memorial
Pass","zip":"20005","lat":"38.9067","long":"-77.0312","short_name":"Alphazap","full_name":"Alphazap Car Dealership","__v":0},{"_id":"66a414825997fc3b7b8ab9c7","id":11,"city":"Carol
Stream","state":"Illinois","address":"8108 Dryden Court","zip":"60351","lat":"41.9166","long":"-88.1208","short_name":"Rank","full_name":"Rank Car Dealership","__v":0},
{"_id":"66a414825997fc3b7b8ab9c8","id":12,"city":"Silver Spring","state":"Maryland","address":"168 Pawling Lane","zip":"20918","lat":"39.144","long":"-77.2076","short_name":"Tin","full_name":"Tin Car
Dealership","__v":0},{"_id":"66a414825997fc3b7b8ab9c9","id":13,"city":"Baltimore","state":"Maryland","address":"452 Fair Oaks Drive","zip":"21275","lat":"39.2847","long":"-76.6205","short_name":"Y-
Solowarm","full_name":"Y-Solowarm Car Dealership","__v":0},{"_id":"66a414825997fc3b7b8ab9ca","id":14,"city":"San Francisco","state":"California","address":"2109 Scott
Parkway","zip":"94147","lat":"37.7848","long":"-122.7278","short_name":"It","full_name":"It Car Dealership","__v":0},{"_id":"66a414825997fc3b7b8ab9cb","id":15,"city":"San
Antonio","state":"Texas","address":"5057 Pankratz Hill","zip":"78225","lat":"29.3875","long":"-98.5245","short_name":"Tempsoft","full_name":"Tempsoft Car Dealership","__v":0},
{"_id":"66a414825997fc3b7b8ab9cc","id":16,"city":"El Paso","state":"Texas","address":"0 Rieder Trail","zip":"79994","lat":"31.6948","long":"-106.3","short_name":"Treeflex","full_name":"Treeflex Car
Dealership","__v":0},{"_id":"66a414825997fc3b7b8ab9cd","id":17,"city":"San Jose","state":"California","address":"7670 American Ash Drive","zip":"95138","lat":"37.2602","long":"-121.7709","short_name":"Home
Ing","full_name":"Home Ing Car Dealership","__v":0},{"_id":"66a414825997fc3b7b8ab9ce","id":18,"city":"Whittier","state":"California","address":"4 Pearson
Avenue","zip":"90605","lat":"33.9413","long":"-118.0356","short_name":"Bitchip","full_name":"Bitchip Car Dealership","__v":0},
{"_id":"66a414825997fc3b7b8ab9cf","id":19,"city":"Hialeah","state":"Florida","address":"93 Monument Circle","zip":"33013","lat":"25.8594","long":"-80.2725","short_name":"Otcom","full_name":"Otcom Car
Dealership","__v":0},{"_id":"66a414825997fc3b7b8ab9d0","id":20,"city":"Detroit","state":"Michigan","address":"4580 Waubesa
Lane","zip":"48224","lat":"42.4098","long":"-82.9441","short_name":"Subin","full_name":"Subin Car Dealership","__v":0},{"_id":"66a414825997fc3b7b8ab9d1","id":21,"city":"San
Francisco","state":"California","address":"046 Mockingbird Junction","zip":"94154","lat":"37.7848","long":"-122.7278","short_name":"Andalax","full_name":"Andalax Car Dealership","__v":0},
{"_id":"66a414825997fc3b7b8ab9d2","id":22,"city":"Fort Lauderdale","state":"Florida","address":"45737 Butternut Lane","zip":"33330","lat":"26.0663","long":"-80.3339","short_name":"Y-Solowarm","full_name":"Y-
Solowarm Car Dealership","__v":0},{"_id":"66a414825997fc3b7b8ab9d3","id":23,"city":"Des Moines","state":"Iowa","address":"21425 Bartelt
Pass","zip":"50936","lat":"41.6727","long":"-93.5722","short_name":"Bitchip","full_name":"Bitchip Car Dealership","__v":0},{"_id":"66a414825997fc3b7b8ab9d4","id":24,"city":"Utica","state":"New
York","address":"408 Delaware Circle","zip":"13505","lat":"43.0872","long":"-75.2603","short_name":"Aerified","full_name":"Aerified Car Dealership","__v":0},
{"_id":"66a414825997fc3b7b8ab9d5","id":25,"city":"Washington","state":"District of Columbia","address":"6505 Melrose
Junction","zip":"20580","lat":"38.8933","long":"-77.0146","short_name":"Opela","full_name":"Opela Car Dealership","__v":0},
{"_id":"66a414825997fc3b7b8ab9d6","id":26,"city":"Pittsburgh","state":"Pennsylvania","address":"306 Jenna Parkway","zip":"15279","lat":"40.4344","long":"-80.0248","short_name":"Flowdesk","full_name":"Flowdesk
Car Dealership","__v":0},{"_id":"66a414825997fc3b7b8ab9d7","id":27,"city":"San Antonio","state":"Texas","address":"95321 Superior
Hill","zip":"78245","lat":"29.4189","long":"-98.6895","short_name":"Namfix","full_name":"Namfix Car Dealership","__v":0},
{"_id":"66a414825997fc3b7b8ab9d8","id":28,"city":"Hialeah","state":"Florida","address":"5458 Maple Way","zip":"33018","lat":"25.9098","long":"-80.3889","short_name":"Fixflex","full_name":"Fixflex Car
Dealership","__v":0},{"_id":"66a414825997fc3b7b8ab9d9","id":29,"city":"San Francisco","state":"California","address":"9 Harper Circle","zip":"94110","lat":"37.7509","long":"-122.4153","short_name":"Fix
San","full_name":"Fix San Car Dealership","__v":0},{"_id":"66a414825997fc3b7b8ab9da","id":30,"city":"Houston","state":"Texas","address":"5423 Spaight
Road","zip":"77218","lat":"29.834","long":"-95.4342","short_name":"Opela","full_name":"Opela Car Dealership","__v":0},{"_id":"66a414825997fc3b7b8ab9db","id":31,"city":"New York City","state":"New
York","address":"5 Northfield Pass","zip":"10131","lat":"40.7808","long":"-73.9772","short_name":"Fintone","full_name":"Fintone Car Dealership","__v":0},
{"_id":"66a414825997fc3b7b8ab9dc","id":32,"city":"Wilkes Barre","state":"Pennsylvania","address":"3 Carey Junction","zip":"18768","lat":"41.2722","long":"-75.8801","short_name":"Subin","full_name":"Subin Car
Dealership","__v":0},{"_id":"66a414825997fc3b7b8ab9dd","id":33,"city":"Des Moines","state":"Iowa","address":"627 Cottonwood Circle","zip":"50335","lat":"41.6727","long":"-93.5722","short_name":"Tres-
Zap","full_name":"Tres-Zap Car Dealership","__v":0},{"_id":"66a414825997fc3b7b8ab9de","id":34,"city":"Silver Spring","state":"Maryland","address":"8 Green
Hill","zip":"20904","lat":"39.0668","long":"-76.9969","short_name":"Gembucket","full_name":"Gembucket Car Dealership","__v":0},
```

Module 3: Deploy sentiment analysis

application deployment verification was successful

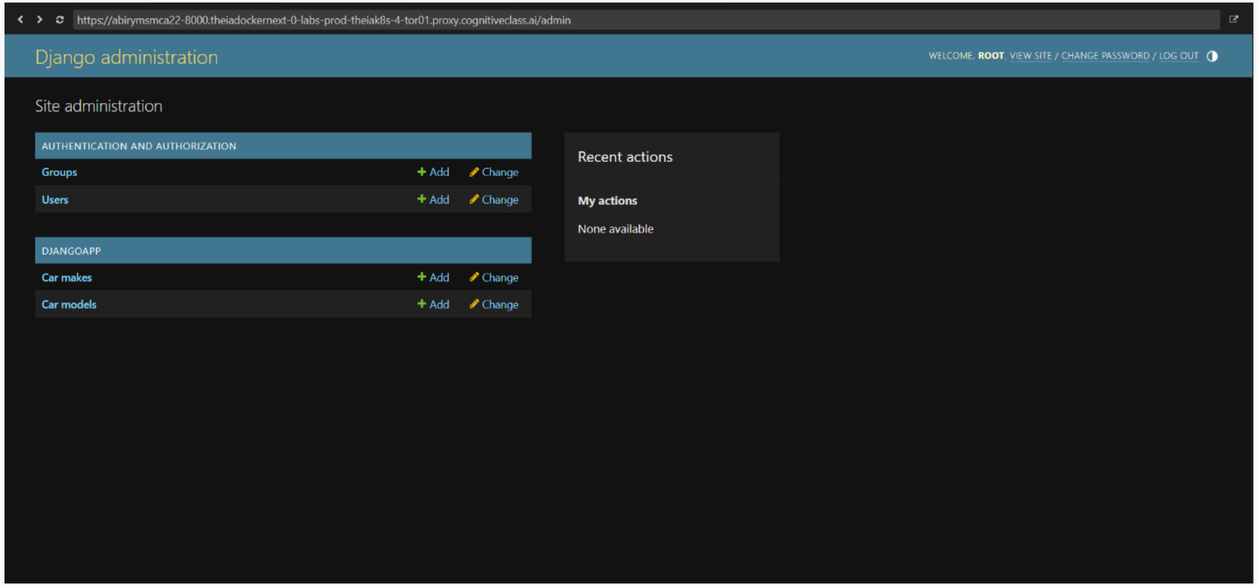


```
https://sentianalyzer.1jzx4ostm7pu.us-south.codeengine.appdomain.cloud/analyze/Fantastic services
```

```
{"sentiment": "positive"}
```

Module 3: admin login

This task verifies the administrator's ability to log in to the Django admin interface. The admin interface is crucial for managing application data, including user accounts and dealership information.



https://abirymsmca22-8000.theiadockernext-0-labs-prod-theiak8s-4-tor01.proxy.cognitiveclass.ai/admin

Django administration

WELCOME, ROOT. VIEW SITE / CHANGE PASSWORD / LOG OUT

Site administration

AUTHENTICATION AND AUTHORIZATION		
Groups	+ Add	Change
Users	+ Add	Change

DJANGOAPP		
Car makes	+ Add	Change
Car models	+ Add	Change

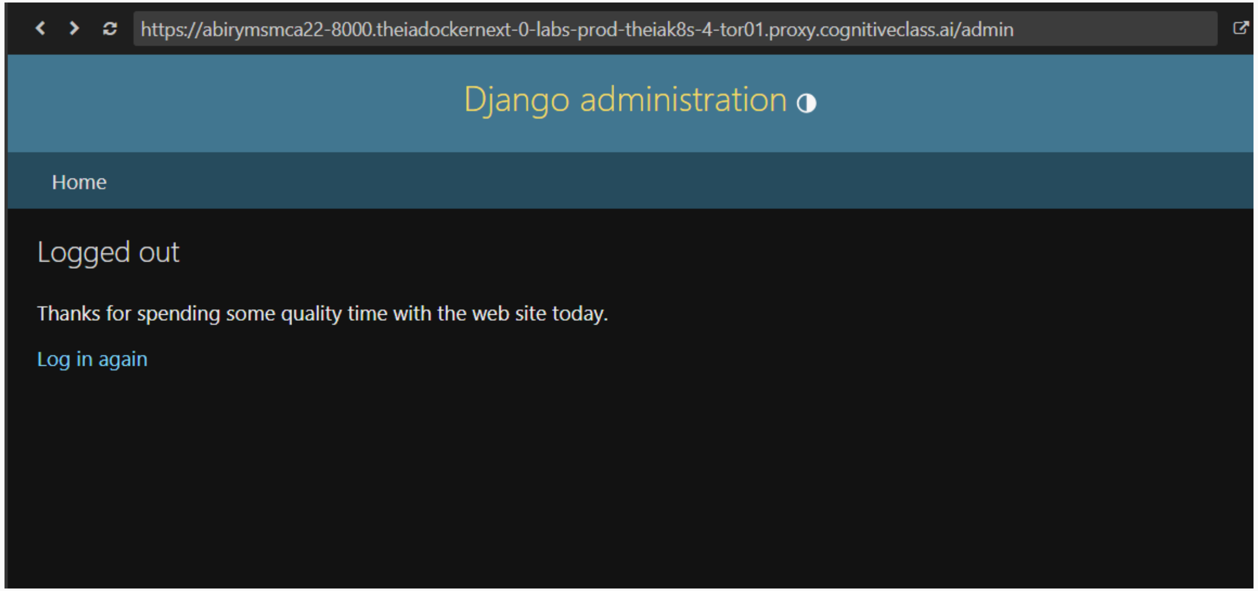
Recent actions

My actions

None available

Module 3: admin logout

This screenshot demonstrates the successful logout process from the Django admin interface, ensuring secure session management for administrators.



https://abirymsmca22-8000.theiadockernext-0-labs-prod-theiak8s-4-tor01.proxy.cognitiveclass.ai/admin

Django administration

Home

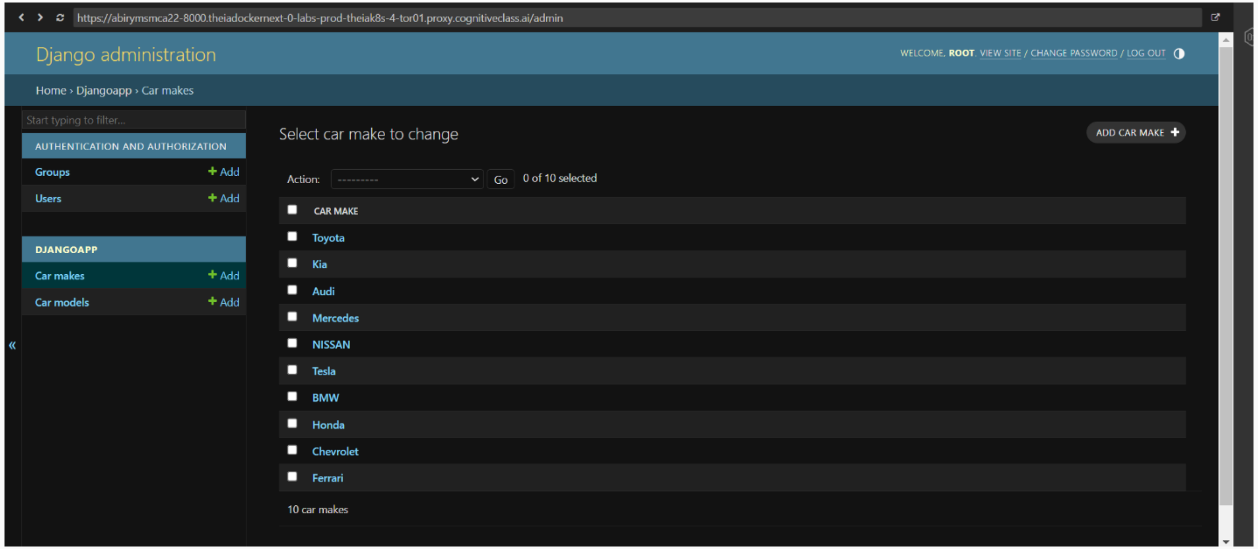
Logged out

Thanks for spending some quality time with the web site today.

Log in again

Module 3: carmake

This task ensures that car makes can be added and viewed in the Django admin interface. It confirms the correct implementation of this feature, allowing administrators to manage the car inventory effectively.



https://abirymsmca22-8000.theiadockernext-0-labs-prod-theiak8s-4-tor01.proxy.cognitiveclass.ai/admin

Django administration

WELCOME, ROOT. VIEW SITE / CHANGE PASSWORD / LOG OUT

Home › Djangoapp › Car makes

AUTHENTICATION AND AUTHORIZATION

Groups	+ Add
Users	+ Add

DJANGOAPP

Car makes	+ Add
Car models	+ Add

Select car make to change

ADD CAR MAKE +

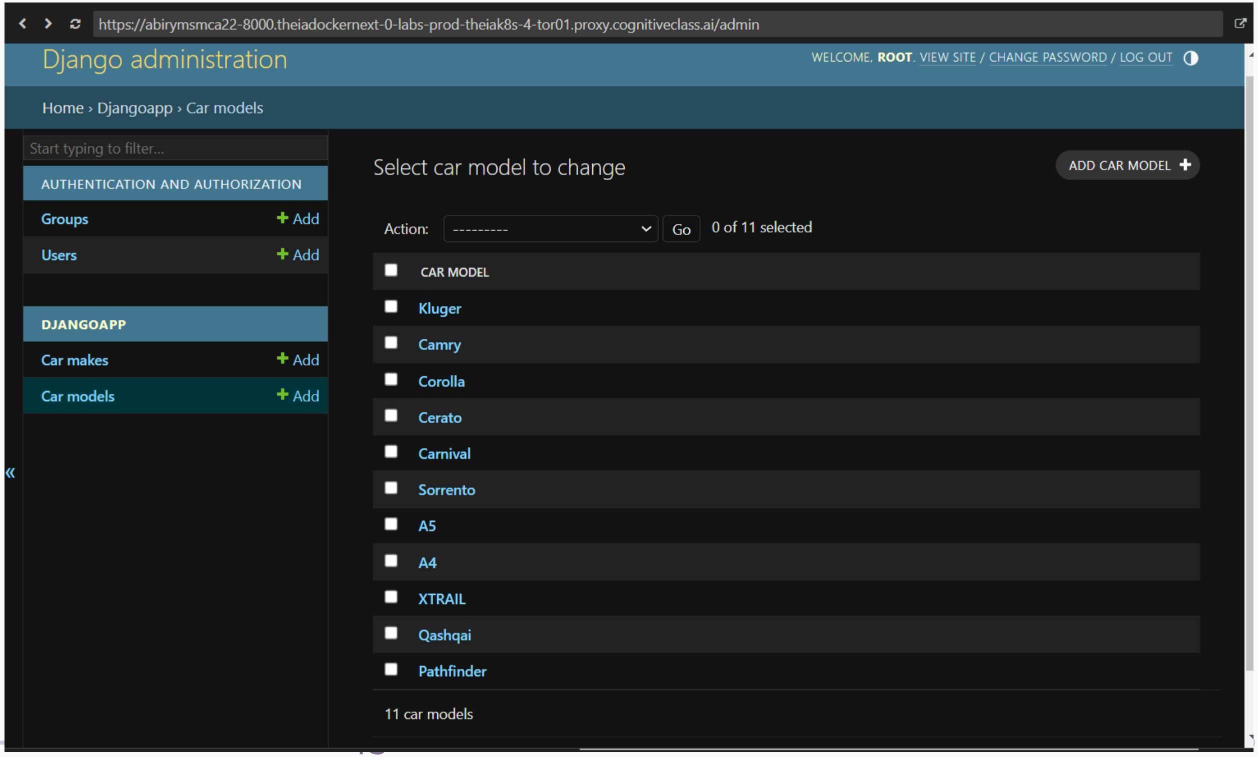
Action: --------- Go 0 of 10 selected

CAR MAKE
Toyota
Kia
Audi
Mercedes
NISSAN
Tesla
BMW
Honda
Chevrolet
Ferrari

10 car makes

Module 3: car model

This screenshot demonstrates the functionality of adding and viewing car models, verifying the relationship between car makes and models. This feature ensures comprehensive management of the car dealership's inventory.



https://abirymsmca22-8000.theiadockernext-0-labs-prod-theiak8s-4-tor01.proxy.cognitiveclass.ai/admin

Django administration

WELCOME, ROOT. VIEW SITE / CHANGE PASSWORD / LOG OUT

Home › Djangoapp › Car models

Select car model to change

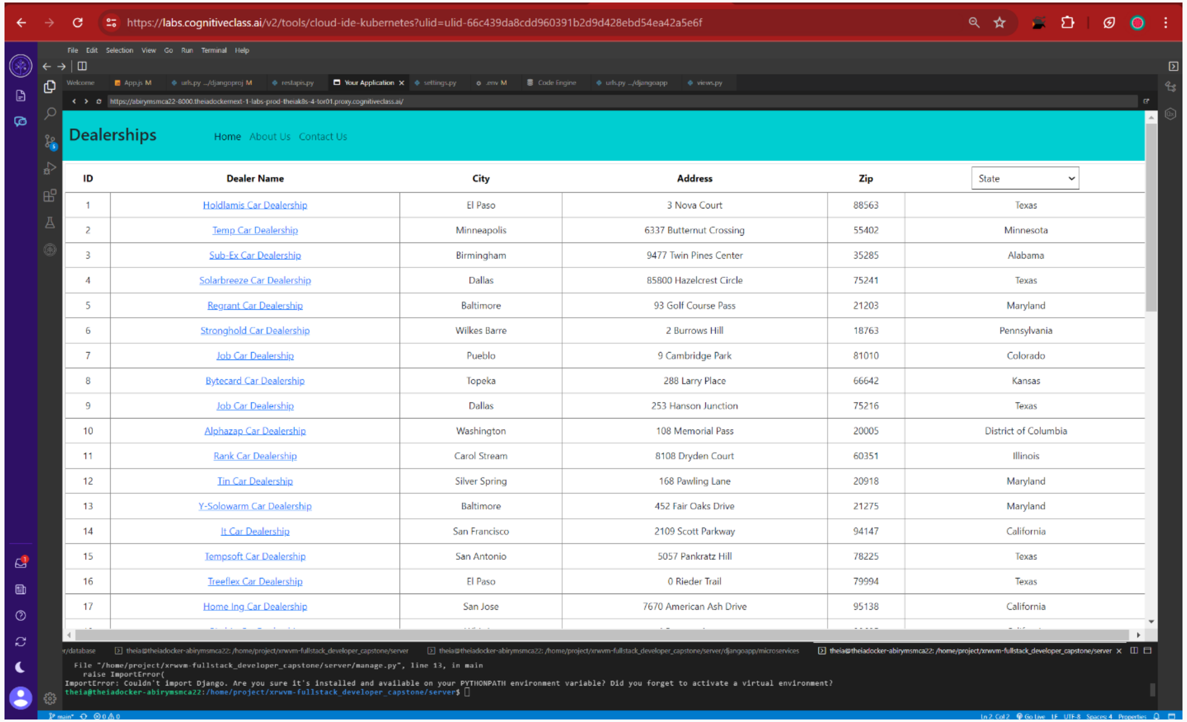
Action: --------- Go 0 of 11 selected

CAR MODEL
Kluger
Camry
Corolla
Cerato
Carnival
Sorrento
A5
A4
XTRAIL
Qashqai
Pathfinder

11 car models

Module 4: dealers on the home page

The screenshot shows the list of dealers visible to unauthenticated users, ensuring that public access to dealer information is properly implemented. This feature provides potential customers with essential dealership information.



https://labs.cognitiveclass.ai/v2/tools/cloud-ide-kubernetes?ulid=ulid-66c439da8cdd960391b2d9d428ebd54ea42a5e6f

https://abirymsmca22-8000.theiadockernext-1-labs-prod-theiak8s-4-tor01.proxy.cognitiveclass.ai/

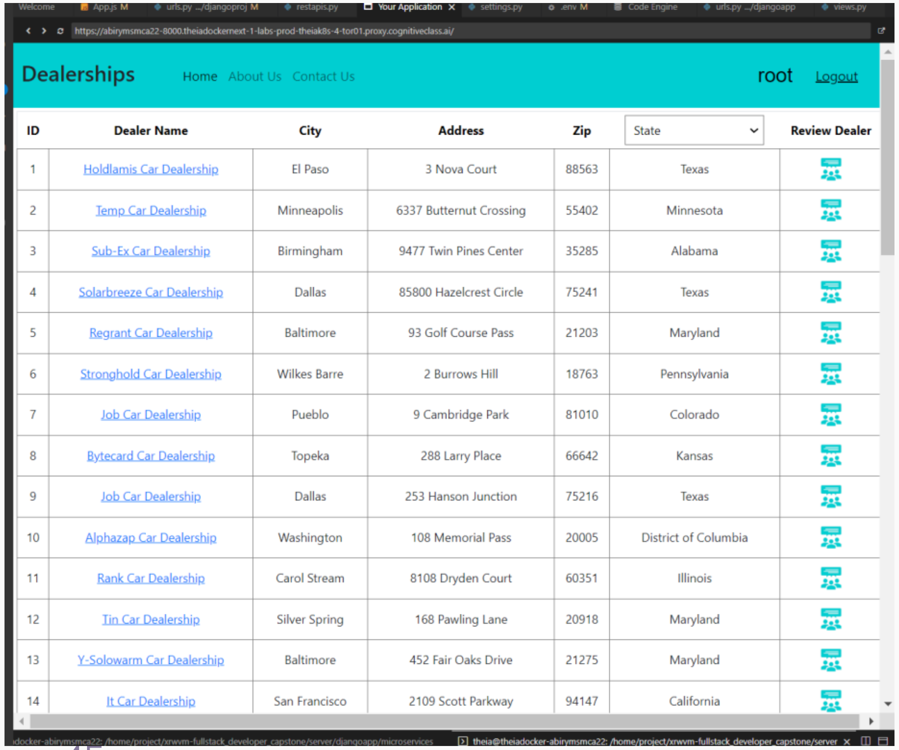
Dealerships Home About Us Contact Us

ID	Dealer Name	City	Address	Zip	State
1	Holdlamis Car Dealership	El Paso	3 Nova Court	88563	Texas
2	Temp Car Dealership	Minneapolis	6337 Butternut Crossing	55402	Minnesota
3	Sub-Ex Car Dealership	Birmingham	9477 Twin Pines Center	35285	Alabama
4	Solarbreeze Car Dealership	Dallas	85800 Hazelcrest Circle	75241	Texas
5	Regrant Car Dealership	Baltimore	93 Golf Course Pass	21203	Maryland
6	Stronghold Car Dealership	Wilkes Barre	2 Burrows Hill	18763	Pennsylvania
7	Job Car Dealership	Pueblo	9 Cambridge Park	81010	Colorado
8	Bytecard Car Dealership	Topeka	288 Larry Place	66642	Kansas
9	Job Car Dealership	Dallas	253 Hanson Junction	75216	Texas
10	Alphazap Car Dealership	Washington	108 Memorial Pass	20005	District of Columbia
11	Rank Car Dealership	Carol Stream	8108 Dryden Court	60351	Illinois
12	Tin Car Dealership	Silver Spring	168 Pawling Lane	20918	Maryland
13	Y-Solowarm Car Dealership	Baltimore	452 Fair Oaks Drive	21275	Maryland
14	It Car Dealership	San Francisco	2109 Scott Parkway	94147	California
15	Tempsoft Car Dealership	San Antonio	5057 Pankratz Hill	78225	Texas
16	Treeflex Car Dealership	El Paso	0 Rieder Trail	79994	Texas
17	Home Ing Car Dealership	San Jose	7670 American Ash Drive	95138	California

```
  File "/home/project/xrwvm-fullstack_developer_capstone/server/manage.py", line 13, in main
    raise ImportError(
ImportError: Couldn't import Django. Are you sure it's installed and available on your PYTHONPATH environment variable? Did you forget to activate a virtual environment?
theia@theiadocker-abirymsmca22:/home/project/xrwvm-fullstack_developer_capstone/server$
```

Module 4: dealers on the home page

Logged-in users can see additional functionality, such as the "Post Review" button, ensuring proper role-based access control. This feature allows authenticated users to contribute reviews.

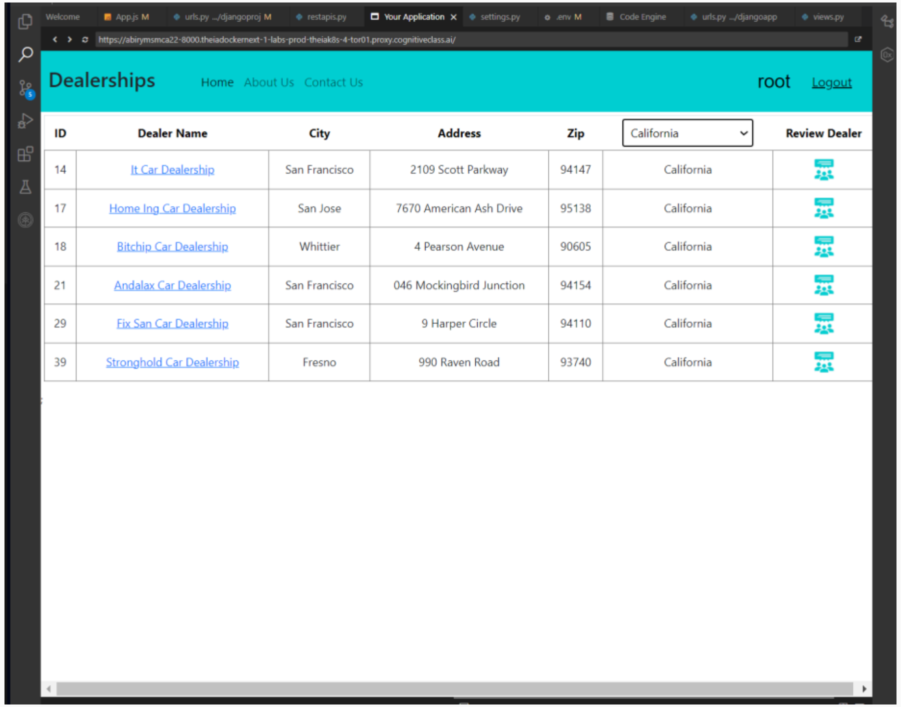


ID	Dealer Name	City	Address	Zip	State	Review Dealer
1	Holdlamis Car Dealership	El Paso	3 Nova Court	88563	Texas	
2	Temp Car Dealership	Minneapolis	6337 Butternut Crossing	55402	Minnesota	
3	Sub-Ex Car Dealership	Birmingham	9477 Twin Pines Center	35285	Alabama	
4	Solarbreeze Car Dealership	Dallas	85800 Hazelcrest Circle	75241	Texas	
5	Regrant Car Dealership	Baltimore	93 Golf Course Pass	21203	Maryland	
6	Stronghold Car Dealership	Wilkes Barre	2 Burrows Hill	18763	Pennsylvania	
7	Job Car Dealership	Pueblo	9 Cambridge Park	81010	Colorado	
8	Bytecard Car Dealership	Topeka	288 Larry Place	66642	Kansas	
9	Job Car Dealership	Dallas	253 Hanson Junction	75216	Texas	
10	Alphazap Car Dealership	Washington	108 Memorial Pass	20005	District of Columbia	
11	Rank Car Dealership	Carol Stream	8108 Dryden Court	60351	Illinois	
12	Tin Car Dealership	Silver Spring	168 Pawling Lane	20918	Maryland	
13	Y-Solowarm Car Dealership	Baltimore	452 Fair Oaks Drive	21275	Maryland	
14	It Car Dealership	San Francisco	2109 Scott Parkway	94147	California	

```
docker-abirymsmca22: /home/project/xrwvm-fullstack_developer_capstone/server/djangoapp/microservices
theia@theiadocker-abirymsmca22: /home/project/xrwvm-fullstack_developer_capstone/server
```

Module 4: dealers by state

This screenshot demonstrates the filtering functionality on the frontend, allowing users to view dealers by state. This enhances user experience by providing customized search results.

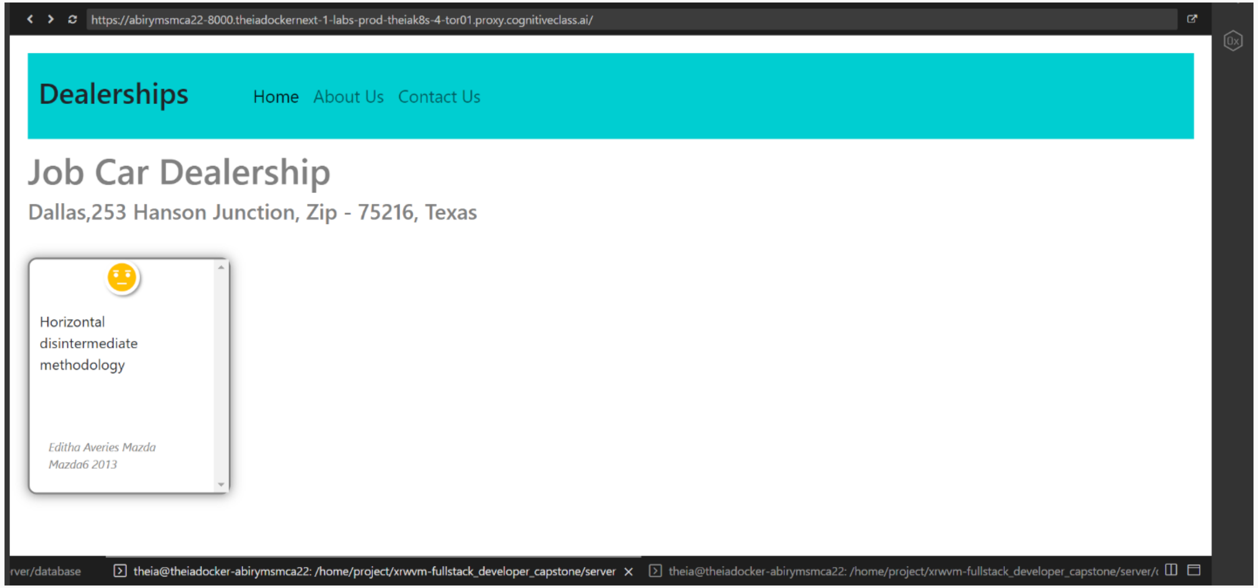


https://abirymsmca22-8000.theiadockernext-1-labs-prod-theiak8s-4-tor01.proxy.cognitiveclass.ai/

ID	Dealer Name	City	Address	Zip	California	Review Dealer
14	It Car Dealership	San Francisco	2109 Scott Parkway	94147	California	
17	Home Ing Car Dealership	San Jose	7670 American Ash Drive	95138	California	
18	Bitchip Car Dealership	Whittier	4 Pearson Avenue	90605	California	
21	Andalax Car Dealership	San Francisco	046 Mockingbird Junction	94154	California	
29	Fix San Car Dealership	San Francisco	9 Harper Circle	94110	California	
39	Stronghold Car Dealership	Fresno	990 Raven Road	93740	California	

Module 4: dealer reviews

This task confirms that users can view detailed dealer information and their reviews, ensuring the front-end correctly displays detailed data. This feature helps users make informed decisions based on dealer reviews.



https://abirymsmca22-8000.theiadockernext-1-labs-prod-theiak8s-4-tor01.proxy.cognitiveclass.ai/

Job Car Dealership

Dallas,253 Hanson Junction, Zip - 75216, Texas

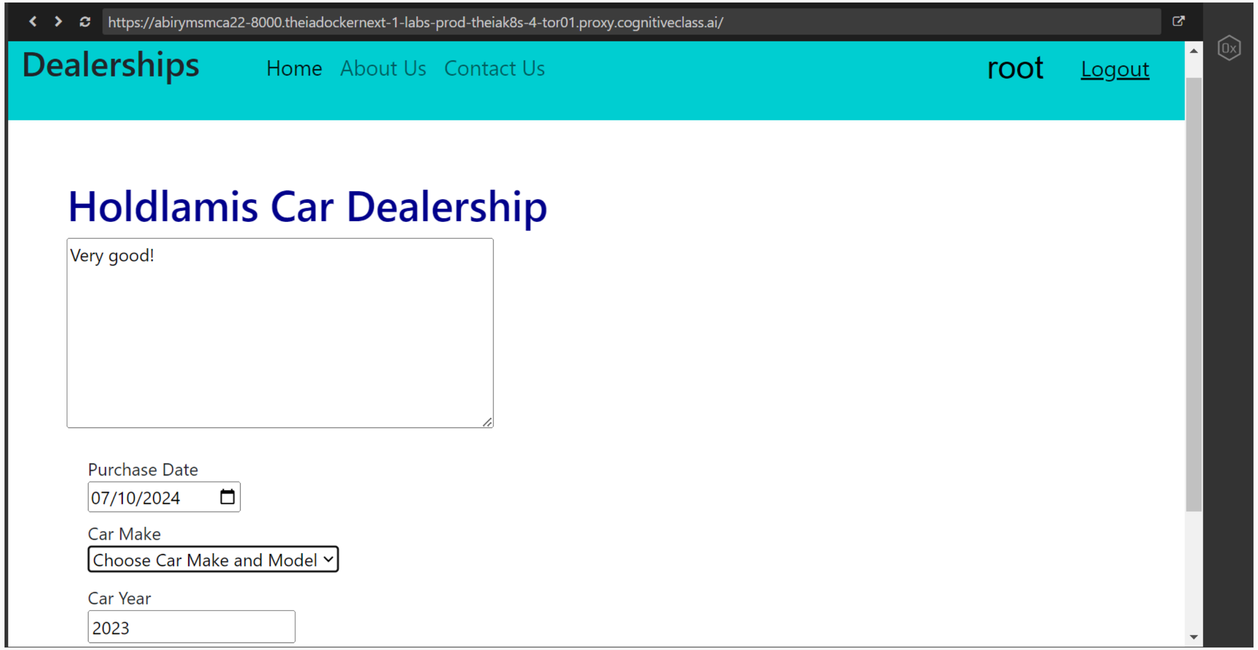
Horizontal disintermediate methodology

Editha Averies Mazda Mazda6 2013

```
rver/database
theia@theiadocker-abirymsmca22: /home/project/xrwvm-fullstack_developer_capstone/server
theia@theiadocker-abirymsmca22: /home/project/xrwvm-fullstack_developer_capstone/server/
```

Module 4: dealer review submission

The screenshot shows the review submission form filled out, ready to be submitted. This ensures the front-end form handling is functioning correctly, allowing users to submit reviews.


https://abirymsmca22-8000.theiadockernext-1-labs-prod-theiak8s-4-tor01.proxy.cognitiveclass.ai/
Dealerships
Home
About Us
Contact Us
root
Logout

Holdlamis Car Dealership

Very good!

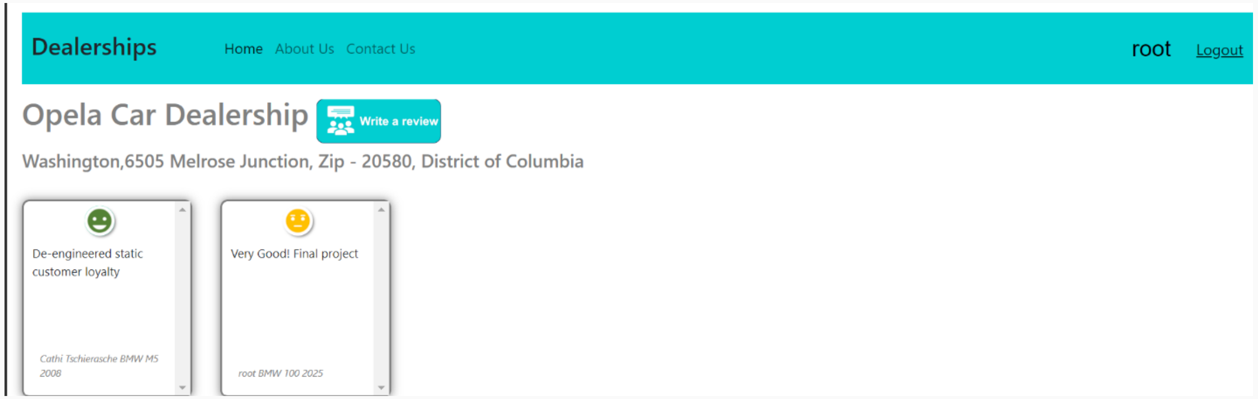
Purchase Date
07/10/2024

Car Make
Choose Car Make and Model

Car Year
2023

Module 4: dealer review submission

This task confirms that reviews are correctly posted and stored in the database, demonstrating the end-to-end functionality of the review feature.



Opela Car Dealership

Washington,6505 Melrose Junction, Zip - 20580, District of Columbia

De-engineered static customer loyalty

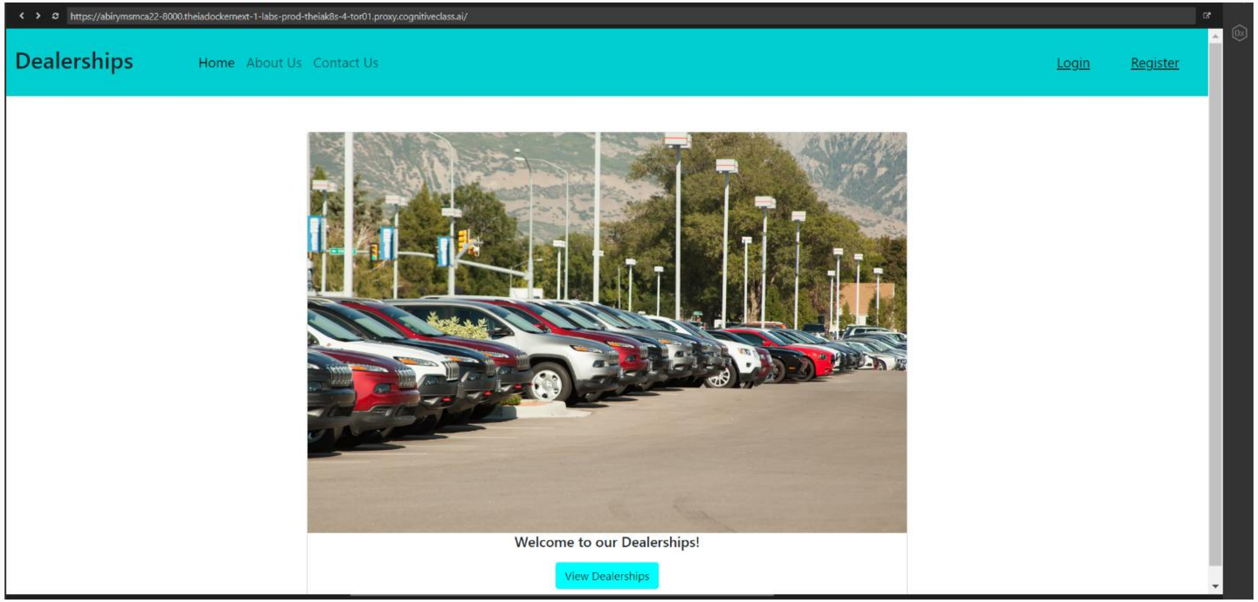
Cathi Tschierasche BMW M5 2008

Very Good! Final project

root BMW 100 2025

Module 5: the deployed application

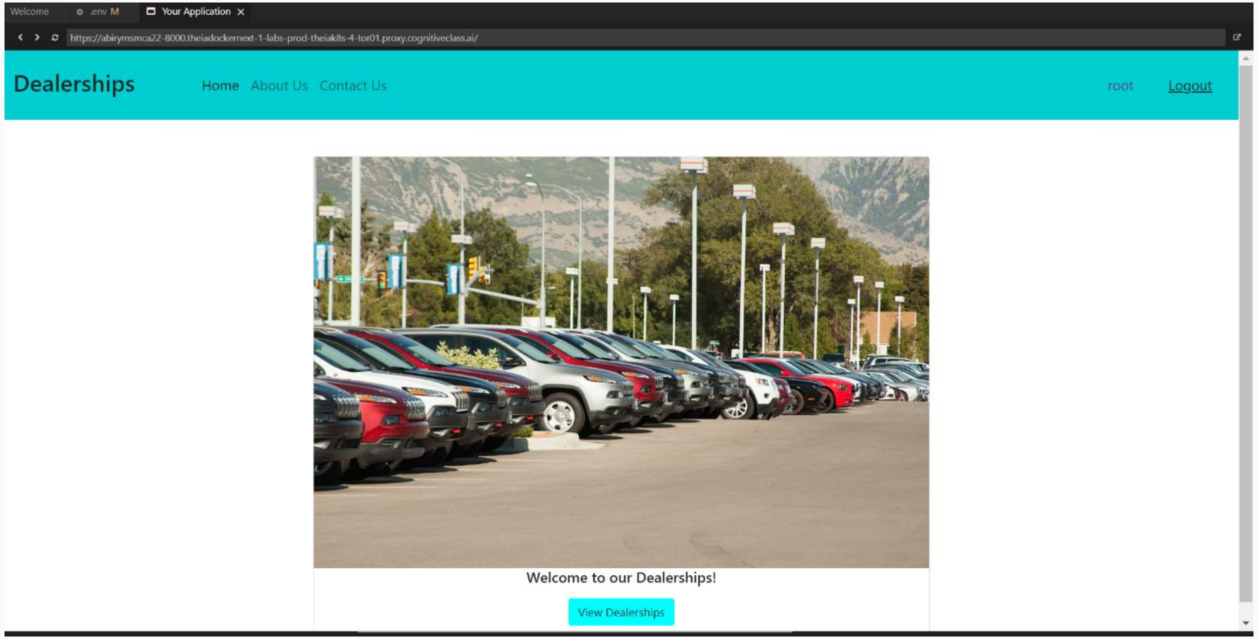
Deployed page
landing page


https://abirymsmca22-8000.theiadockernext-1-labs-prod-theiak8s-4-tor01.proxy.cognitiveclass.ai/

Welcome to our Dealerships!

Module 5: the deployed application

Deployed page logged in

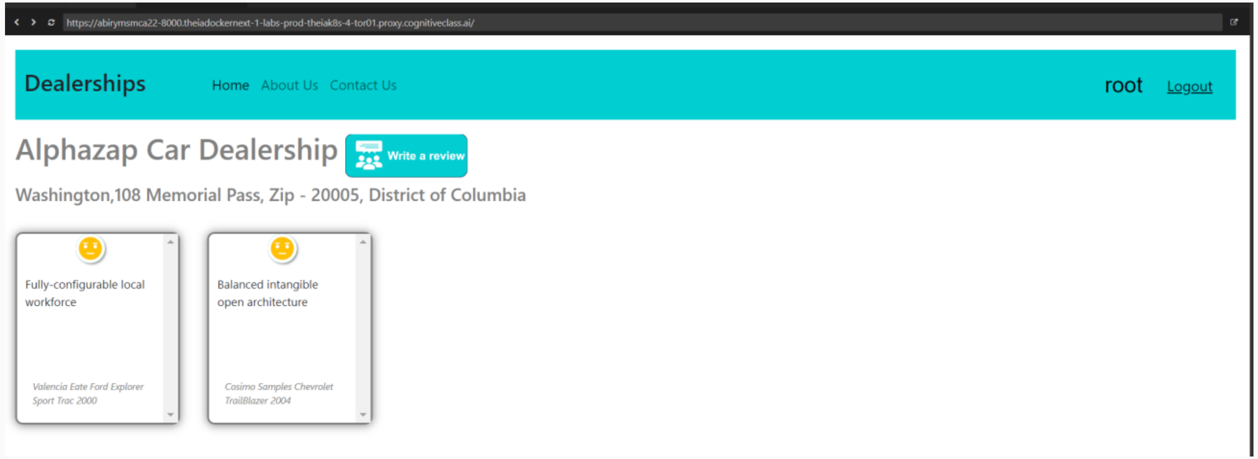

Welcome
.env M
Your Application
https://abirymsmca22-8000.theiadockernext-1-labs-prod-theiak8s-4-tor01.proxy.cognitiveclass.ai/
Dealerships
Home
About Us
Contact Us
root
Logout
View Dealerships

Welcome to our Dealerships!

Module 5: the deployed application

Deployed page

Dealer details


https://abirymsmca22-8000.theiadockernext-1-labs-prod-theiak8s-4-tor01.proxy.cognitiveclass.ai/

Alphazap Car Dealership

Washington,108 Memorial Pass, Zip - 20005, District of Columbia

Fully-configurable local workforce

Valencia Eate Ford Explorer Sport Trac 2000

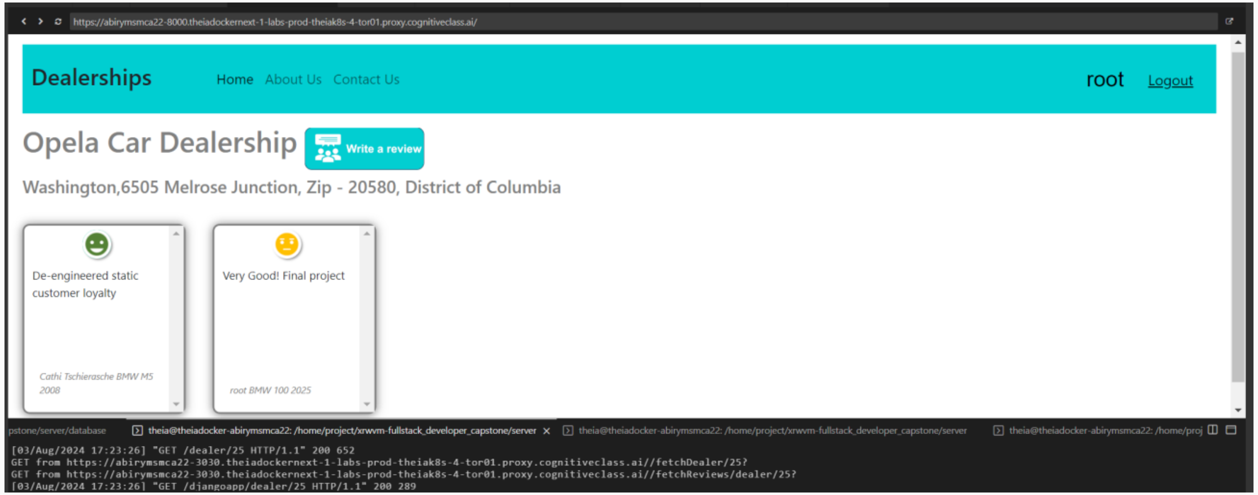
Balanced intangible open architecture

Cosimo Samples Chevrolet TrailBlazer 2004

Module 5: the deployed application

Deployed page

Add new review


https://abirymsmca22-8000.theiadockernext-1-labs-prod-theiak8s-4-tor01.proxy.cognitiveclass.ai/
Dealerships
Home
About Us
Contact Us
root
Logout
Write a review

Opela Car Dealership

Washington,6505 Melrose Junction, Zip - 20580, District of Columbia

De-engineered static customer loyalty

Cathi Tschierasche BMW M5 2008

Very Good! Final project

root BMW 100 2025

pstone/server/database
theia@theiadocker-abirymsmca22: /home/project/xrwvm-fullstack_developer_capstone/server
theia@theiadocker-abirymsmca22: /home/project/xrwvm-fullstack_developer_capstone/server
theia@theiadocker-abirymsmca22: /home/proj

```
[03/Aug/2024 17:23:26] "GET /dealer/25 HTTP/1.1" 200 652
GET from https://abirymsmca22-3030.theiadockernext-1-labs-prod-theiak8s-4-tor01.proxy.cognitiveclass.ai//fetchDealer/25?
GET from https://abirymsmca22-3030.theiadockernext-1-labs-prod-theiak8s-4-tor01.proxy.cognitiveclass.ai//fetchReviews/dealer/25?
[03/Aug/2024 17:23:26] "GET /djangoapp/dealer/25 HTTP/1.1" 200 289
```

09 Challenges

Challenges faced

1. IBM Cloud Engine Issues: Integrating and deploying the application on the IBM Cloud Engine presented challenges, possibly due to configuration complexities or platform-specific constraints.
2. Rhyme Environment Performance: The project was developed in the Rhyme environment provided by Coursera, which occasionally experienced slow performance. This lag led to inefficiencies, as tasks often had to be repeated, resulting in a loss of time and effort.
3. Testing and Debugging: Difficulties in testing individual components due to interdependencies, especially in a full-stack environment. Debugging was challenging due to complex error tracing across multiple layers of the stack.
4. Deployment Issues: Challenges related to deploying the application on IBM Cloud, including managing environment configurations and ensuring the application runs smoothly in the cloud environment.
5. Time Management: Managing time effectively to complete the project milestones within the set deadlines while maintaining code quality and meeting the project requirements.

10 Conclusion

Conclusion

The Car Dealership Review App project provided hands-on experience in full-stack development, from front-end and back-end integration to cloud deployment. Despite challenges like integration complexities and performance optimization, the project was completed successfully, resulting in a functional and user-friendly application. This experience strengthened my technical skills and prepared me for real-world full-stack development.

11 References

References

- Django: https://docs.djangoproject.com/en/stable/
- Express.js: https://expressjs.com/en/4x/api.html
- MongoDB https://docs.mongodb.com/manual/
- Kubernetes : https://kubernetes.io/docs/home/
- Github actions: https://docs.github.com/en/actions
- Python: https://docs.python.org/3.12/
- React: https://reactjs.org/docs/getting-started.html

Thanks!