Add Static pages

Estimated time needed: 1 hour

Congratulations on your new role as the Full Stack Application Developer at `Best Cars` dealership.

As part of the project, your first task is to run and test its main Django application. You are given a skeleton of the Django application as a starting point.

Then, you need to complete the following steps to add additional static pages.

> The lab environment doesn't store your changes. If you are going to leave your workspace in between, it is highly recommended that you push any changes you make to git.

▼ Click here for the procedure to push your changes into Git repo

```
1. 1
1. git config --global user.email "yourgithub@email.com"
```

Copied!

```
1. 1
1. git config --global user.name "name"
```

Copied!

```
1. 1
1. git add .
```

Copied!

```
1. 1
1. git commit -m"Adding temporary changes to Github"
```

Copied!

```
1. 1
1. git push
```

Copied!

Fork and Clone the repository

1. Navigate to this link: link and create a fork of the repository that includes essential starter code required for this project.
2. Go to your forked repository.
3. Copy the git clone URL of the skeleton repository that you forked into your account.

   Make sure you copy your repository by checking the URL in the address bar.
4. Open a new terminal.
5. Clone the URL you copied in step 3.

```
1. 1
1. git clone <your_repo_name>
```

Copied!

Run the Django app on development server

1. Observe the folder structure of the Django app skeleton structure. You will see `server` folder with three sub-folders:

- `djangoapp`: Contains the django application
- `djangoproj`: Contains the project configuration
- `frontend`: HTML and CSS and React front end

Next, let's setup the Python runtime in Theia and test the app.

2. Change to the `server` directory in the terminal.

```
1
```

```
cd xrwvm-fullstack_developer_capstone/server
```

Copied!

3. Set up virtual environment for your Django application to run in.

```
1
2
3
```

```
pip install virtualenv
virtualenv djangoenv
source djangoenv/bin/activate
```

Copied! Executed!

4. Install the necessary Python packages in your virtual environment. The package names have already been provided in `requirements.txt`.

```
1
```

```
python3 -m pip install -U -r requirements.txt
```

Copied!

5. In `server/djangoproj/settings.py`, under `TEMPLATES`, you will find `DIRS` as an empty list. Add the `os.path.join(BASE_DIR,'frontend/static')` to the list for the Django application to recognize the front-end static files. It should be set up as below.

```
1
2
3
```

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

Copied!

6. In the same file, `server/djangoproj/settings.py`, add the directory for the Django application to look for static files at the bottom of the file.

```
1
2
3
```

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

Copied!

7. Perform migrations to create necessary tables.

```
1
```

```
python3 manage.py makemigrations
```

Copied!

8. Run migration to activate models for the app.

```
1
```

```
python3 manage.py migrate
```

Copied!

9. Start the local development server.

```
1
```

```
python3 manage.py runserver
```

Copied!

Great! You should have the Django app running now.

10. Please take a screenshot of the console or terminal output to show that your app is running successfully, and store it as `django_server.png` or `django_server.jpg` for peer review.

11. Click the `Django Application` button below to open the application. You will see the static homepage that is rendered. Copy the URL from the browser.

    The links in the page will not work as yet.

Django Application

▶ If the page doesn't open, click here to open the URL manually.

---

10. On the file editor, open the `server/djangoproj/settings.py` in the editor and set the `ALLOWED_HOSTS` and `CSRF_TRUSTED_ORIGINS` to reflect your Django app's root URL.

    Please do not included the `/` at the end.

```
1
2
```

```
ALLOWED_HOSTS=['localhost','<your application URL here>']
CSRF_TRUSTED_ORIGINS=['<your application URL here>']
```

Copied!

Add 'About Us' page

1. Open `server/frontend/static/About.html` in the editor.
2. You have been provided with the stylesheet `style.css` in the same folder. In the `<head>` tag inside `About.html` link the stylesheet for use in the HTML file.

```
1
2
```

```
  <link rel="stylesheet" href="/static/style.css">
  <link rel="stylesheet" href="/static/bootstrap.min.css">
```

Copied!

3. Paste the following content in the `<div>` tag named `about-header`.

```
1
2
```

```
  <h1>About Us</h1>
  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.
```

Copied!

   You can add additional information that might be relevant to this page.

4. Change the image `person.png` to an actual person's image and change all the text in `About.html` to look more realistic. Change the styling as per your preference.
5. Go to `djangoproj/urls.py` and add the following to the `urlpatterns`.

```
1
```

```
    path('about/', TemplateView.as_view(template_name="About.html")),
```

Copied!

6. The Django server automatically restarts. Click on the button below and check if the changes are reflected.

Dealership Application

7. Take a screenshot of the page and store it as `about_us.png` or `about_us.jpg`. Ensure that the URL in the address bar is visible in the screenshot.

Add 'Contact Us' page

1. Under `server/frontend/static` folder, add a new file named `Contact.html`.
2. Add the style sheets link.

3. Add the header navigation bar, with `Contact Us` as the active link.
4. Write the contact information content in the file. You can be creative and make up your information. Use css to style it.
5. Add the `contact` path to `djangproj/urls.py`.

```
1
    path('contact/', TemplateView.as_view(template_name="Contact.html")),
```

Copied!

6. The Django server automatically restarts. Click on the button below and check if the `contact` page renders.

Dealership Application

7. Take a screenshot of the page and store it as `contact_us.png` or `contact_us.jpg`. Ensure that the URL in the address bar is visible in the screenshot.

Submission

Please note down the URLs of your GitHub repo and take screenshots of the static pages for peer reviews.

Summary

In this lab, you prepared your GitHub repository and cloned an app skeleton to start building the dealer review app. As a warm-up task, you have created several static pages and tested the app. Please push the changes you made into your GitHub repository.

Now, you are ready to start some actual design and development work.

Author(s)

Lavanya