Exam Preparation

Problems for the exam preparation for the "Software Engineering and DevOps" course @ SoftUni.

You are given a simple Node.js application with failing tests. You should work in GitHub, fix the failing tests and deploy the app to Render.com.

1. Resource Files

You are provided with a JS application – download it from the resources for the exam preparation session.

You can test the application locally using the commands, described in the **`package.json`** file.

Don't forget to use **`npm install`** command.

2. Working with Git and GitHub (25 points)

Create a New Repository in GitHub

Create a new repository in GitHub and upload the project files there. The repository should be named **`{your-softuni-username-exam-prep}`**.

Clone the Repository Locally

Clone the repository to your local machine.

Add, Commit and Push the Resource Files

Add the project resource files and commit them to the repository. Push the changes to the repo.

Create a Junior Developer Branch

In the local environment, navigate to the directory of the cloned project. Create a new branch and switch to it. The name of the branch should be **`junior-dev`**.

Fix the Failing Tests

After you have created the **`junior-dev`** branch, please refer to point 3. Fix Failing Tests. Once you have fixed the tests, proceed with the next steps.

Commit and Push Changes to New Branch

Without making any changes yet, push the **`junior-dev`** branch to GitHub.

Create Pull Request

Now, go to GitHub and create a pull request to propose merging the **`junior-dev`** branch into the **`main`** branch. Finally, merge the two branches.

NOTE: Do not delete the `junior  dev` branch. You can delete it or remove the entire repository AFTER you receive your exam grade!

3. Fix Failing Tests (25 points)

The application has some integration and UI tests and some of the UI tests are failing. Your task is to fix the UI tests and be sure that all of them pass. The app is using the **Playwright** framework.

4. Implement CI (25 points)

You should implement a CI system in GitHub actions to build and test the application.

Add a custom step to the build script in order to display your softuni username.

NOTE: Remove the UI tests from the build script as the CI process won't be successful.

5. Implement CD (25 points)

Using the GitHub action script, implement a CD workflow to auto deploy the app in **Render.com.**

Modify the domain in Render.com to visualize your Softuni username.

6. Add UI Tests to CI/CD

After you have successfully ran the CI/CD workflow and your app has been deployed to Render.com, add the UI tests to your GitHub repository and to the CI/CD workflow.

NOTE: Don't forget to change the URL in the tests so that it corresponds to your deployed application.

7. Requirements

Provide the **GitHub** repository and the **Render.com links** in a **Word** document**.**

Provide an image from the **GitHub Actions** result from **GitHub.**

8. Submission

You can submit your solution in **one** of the following ways **(chose whichever is more convenient for you):**

- Add the **Word document (containing the two links)** and the **image** in **one .ZIP** file and **submit** it to the **SULS** system;
- **Paste** the **image** in a **Word** document and add both of the links in it. Add the **.docx** file to **one .ZIP** file and **submit** it to the **SULS** system.