

# Software Engineering & DevOps Exam – 16 December 2023

Problems for the regular exam 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 section.

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}**.

### 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.

### Make Changes to the Readme file

In the **main** branch, alter the **Readme** file to show your username.

### 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. Review the pull request and 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

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

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

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. It should be something like this {your-softuni-username-exam}**

### 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 by filling the provided document resource with name "**Solution-Resources-Template-Software-Engineering-and-DevOps**".

You should fill in the following information:

- The link for your GitHub repository;
- The link for your Render.com service;
- A screenshot of the workflow in GitHub Actions.

After you have filled in the document, zip it and upload it to the SALS system.

**NOTE: Don't forget that you must NOT delete the junior-dev branch, nor the GitHub repo, not the Render.com service until you receive your exam grade!**