Part 1 (max. 1.5 p). Design

(a) Overview (max. 0.5 p)

Give a short overview as text describing your topic and the idea behind the static and the dynamic parts.

My idea is to create a personal portfolio website that shows my skills, experience, and professional background as a web development trainee. The website aims to present both my personal information and your achievements, while also providing a way for potential employers or collaborators to reach out to me.

1. **Static parts**

1.1 HTML Structure: The core structure of the pages like the header, footer, and sections like About, Skills, and Projects are static. These sections contain fixed content such as:

- My personal information (name, status, field of study).
- Academic achievements (e.g., GPA).
- Professional goals and personal qualities.
- The list of skills (both programming and soft skills) with icons and text.
- The project titles and descriptions.
- Social media and contact information in the footer.
- Static images like your profile picture.
- Links to external platforms (e.g., GitHub, LinkedIn, Facebook).

1.2 CSS Styles: The design layout (fonts, colors, positioning) is handled in the style.css file, providing a consistent look and feel for each page. This part is static as it doesn't change unless manually updated.

1.3 Icons and Links: The use of icons from Boxicons for visual appeal in sections like skills and the footer is static. The navigation menu is also static, allowing access to other pages like About, Skills, and Projects.

2. **Dynamic Parts**

2.1 Contact Form: The contact form functionality is dynamic, where users can input their details (name, email, message) and submit the form. This triggers JavaScript functions to:

- Show and hide the form (openContactForm and closeContactForm).
- Display an overlay when the form is open.
- Handle the form submission by preventing the default behavior and displaying a confirmation message ("Thank you for contacting me!").

2.2 Notification Counter: When a user submits the contact form, a notification counter dynamically updates the number of contacts submitted. This is done by modifying the DOM with JavaScript after form submission. The counter is shown in the notification icon at the top.

2.3 Button Interactions: The "Hire me" button triggers the contact form display, which is dynamic as it responds to user interaction (clicking the button).

(b) Structure (max. 1 p)

Explain the structure and navigation of your portfolio. You can use a wireframe or other visualization (applying a tool such as Draw.io or drawing by hand and scanning or taking a picture) or text. Also, explain the file and directory structure used.

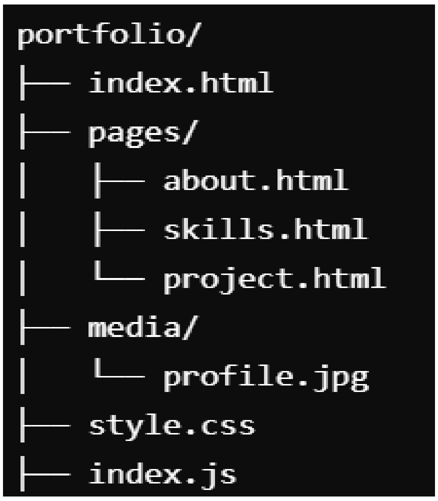
1. Structure:

- **Home Page: Introduction, achievements, "Hire me" button, profile picture.**
- **About Page: Personal details, academic achievements, professional goals.**
- **Skills Page: Technical and soft skills, with icons.**
- **Projects Page: List of completed projects with descriptions and GitHub links.**
- **Footer: Contact info and social media links.**
- **Contact Form: Opens from "Hire me" button.**

2. Navigation:

- **Header: Contains links to Home, About, Skills, and Projects.**
- **Main Section: Displays intro, achievements, and button to contact form.**
- **Footer: Contact info and social media links.**

3. File & Directory Structure:



```
portfolio/
├── index.html
├── pages/
|    ├── about.html
|    ├── skills.html
|    └── project.html
├── media/
|    └── profile.jpg
├── style.css
├── index.js
```

Part 3 (max. 2.5 p). Implementation of the dynamic parts

(a) Description (max. 0.5 p)

Describe here in more detail the dynamic parts you have mentioned in your part 1. This means that you should give an overview of the JavaScript used (such as functions and their use) as well as explain the events that dynamic functionality is based on. Remember to meet the requirements stated in section Portfolio.

Contact Form Modal:

- **Functions**:
  - openContactForm(): This function makes the contact form visible by changing the display style to flex and shows the overlay. It is triggered when the "Hire me" button is clicked.
  - closeContactForm(): This function hides the form and overlay by setting their display style to none. It is triggered when the close icon (bx-x) is clicked.
- **Event Handling**:
  - The form's submit event is intercepted using form.addEventListener("submit", function (event) {...}). This prevents the default form submission, closes the form, and shows an alert thanking the user for contacting. The event also updates a counter of the number of form submissions.

Notification Icon:

- A notification counter (numOfContactForms) is dynamically updated every time the form is successfully submitted.
- When a form is submitted, the counter increments and the value is displayed next to the notification bell icon. This is done by manipulating the text content of the dynamically created numOfContactForms paragraph element.