

CS 463 Personal Website Documentation

[Project Overview](#)

[Description](#)

[Project Initialization](#)

[Getting Started](#)

[Making Changes and Contributing](#)

[Deployment \(on GitHub Pages\)](#)

[Code Format and Structure](#)

[File Structure](#)

[Code Formatting](#)

[Code Documentation \(src\)](#)

[index.jsx](#)

[routes.jsx](#)

[components](#)

[Navbar](#)

[SocialMedia](#)

[pages](#)

[AboutMe](#)

[Contact](#)

[Previous Work](#)

[Projects](#)

[styles](#)

[External Resources Used](#)

Project Overview

Description

This project is Timofey Shichalin's final project for the class CS 463 (Introduction to Web Development) at Portland State University (winter 2023). The project is a multi-page website that introduces Timofey Shichalin professionally and displays his past experience (education, skills, projects, etc). The project is deployed and can be viewed here: <https://timshicha.github.io/cs463-personal-website/>.

Project Initialization

This project was initialized using `npx create-react-app`.

Getting Started

Running this project locally requires some setting up. You may follow the steps below to get this project running:

- 1) Install Git and Node on your machine.
Git download: <https://git-scm.com/>
Node download: <https://nodejs.org/en/download/>.
- 2) Clone the repository. In Powershell (or an equivalent terminal), enter `git clone https://github.com/timshicha/cs463-personal-website` to clone the repository in the current directory.
- 3) Enter into the cloned repository using `cd cs463-personal-website` and run `npm install`. This will install all of the required dependencies to run the app, creating a `node_modules` directory.
- 4) Run the project locally using `npm run start`. This will start the project locally. You can view the project on (<http://localhost:3000>).

Making Changes and Contributing

Once you've started running the project using `npm run start`, any changes you make to the project will be automatically reflected in the webpage on `localhost`.

Deployment (on GitHub Pages)

If you want to deploy the website on your own GitHub page, create your own git remote repository. Prior to deployment, open the `package.json` file and configure the `homepage` variable (line 4) to the domain you'll be deploying the project on. If you'll be deploying on GitHub pages, your domain will be

`https://<YOUR_GITHUB_USERNAME>.github.io/<YOUR_REPOSITORY_NAME>/`.

Then, run the GitHub deployment script using `npm run deploy` (or `npm run deploy -m "message"`). This will create an optimized build of the project and push it to the `gh-pages` branch of your repository. When setting up the project to run from GitHub pages, select the project to deploy the builds from the branch `gh-pages (root)`. This will automatically redeploy when new builds are pushed to the `gh-pages` branch.

Code Format and Structure

File Structure

The project uses the following file structure (important files are included):

- `public`
 - `index.html` (can access files only in the same directory using `%public%`)
- `src`
 - `assets` (holds all imported content [like images] that is used in the project)
 - `components` (holds components that can be reused by many pages)
 - `pages` (holds the pages; the React router in `src/routes.jsx` uses these pages)
 - `styles` (holds the stylesheets used in the project)

Code Formatting

The code was auto-formatted using the Prettier extension in VSCode.

- Indentation: 2 spaces
- Quotation rules:
 - JS/jsx code: single quotes
 - HTML code embedded within JS/jsx: double quotes
 - CSS: single quotes

JSX code (enclosed in curly braces) may appear inside the embedded HTML code. This code is also to be in single quotes.

Code Documentation (src)

`index.jsx`

This file renders the app content in the `index.html` file.

`routes.jsx`

This file is responsible for rendering the proper component (that is, pages) when the user's path changes. More pages can be added using the `<Route>` tag, with the attribute `path` specifying the user's path and the attribute `element` specifying the component (or page) to render.

components

Navbar

The `Navbar` component places a navigation bar at the top of the page. Really, the `Navbar` is two navigation bars: the first one is the horizontal navigation bar you see on wide screens, and the second is the vertical expandable navbar that appears on narrower (or mobile) screens. Depending on the width size (see media queries in `src/styles/Navbar.css`), one of the Navbars will be visible while the other one is hidden.

`menuExpanded`

This variable only affects the appearance of the page when the screen is narrow and the expandable navigation bar is present. The expanded navigation bar list will only be visible when `menuExpanded` is set to `true`. To update `menuExpanded`, use the `setMenuExpanded` function. This function is already called when the user clicks the hamburger icon (three horizontal lines icon).

SocialMedia

The `SocialMedia` component is a div with two icons: LinkedIn and GitHub. Clicking on them redirects the user to Timofey's LinkedIn or GitHub profile.

The images were downloaded from <https://icons.getbootstrap.com/>.

pages

AboutMe

The `AboutMe` page consists of a grid that changes in structure when the width of the window changes. On wider screens, the grid is composed of two columns (text area and photo/social area). On narrower screens, the grid is one column (two rows). The photo/social area is directly above the text area.

Contact

The `Contact` page allows the user to create a message and simulate sending it (it's actually non-functional). When the user submits, the fields they entered are examined. If any of them is empty, an alert message is displayed. If they properly fill out the form, it gets printed in the console and the user gets redirected back to the home (About Me) page.

Previous Work

The `PreviousWork` page displays some of Timofey's skills and accomplishments. The page contains expandable sections. A maximum of one section can be expanded at any time. CSS transitions allow the sections to expand and collapse smoothly (see `src/styles/PreviousWork.css`).

`expanded`

This variable holds the name of the section that's currently expanded (or `null`).

When a user clicks the expand button on any of the sections, this variable is updated using `setExpanded`. Each of the expanded sections has a `className` attribute that adds the class `expanded-sz` when the section is the one that should be expanded. This class changes the expanded section's height from `0px` to `150px`, allowing the content to be visible.

Projects

The `Projects` page displays a few projects. Each project is displayed using a grid. On wider screens, the image is displayed alongside the project descriptions. On narrower screens, the Project description is displayed below the project image. For more specific details, see the media queries in `src/styles/Projects.css`.

styles

The stylesheets are stored in the `styles` directory. Each page and component has its own style sheet for styles that are specific to that page or component. Styles that are used across the app (in more than one component) are placed in `styles/GeneralStyels.css`.

External Resources Used

- <https://developer.mozilla.org/en-US/docs/Web/CSS>
- <https://icons.getbootstrap.com/>