
Lab 6: Working with Front-End Frameworks [React]

Learning Outcomes:

- Familiarize yourself with React using Create React App.
- Understand the general structure of the framework
- Work individually to create a simple React App.
- Deploy the react app on Netlify.

Instructions:

- For this lab, you will be creating a basic React app using node.js and ‘create-react-app’.
- Make sure node.js and npm are installed on your system.
- Create a new app with this command: ‘npx create-react-app <app_name>’
- NPM will create a new app. To run the app and check UI, run ‘npm start’ inside the app folder.
- Go through the app folder and file structure.
- Modify the boilerplate code in ‘App.js’ a bit as per your requirement (Show off your creativity)
- Deploy the app on Heroku (follow instructions below).

Note: Remember, you’ll be expected to use React concepts to build the application.

- In regard to the look-and-feel of your assignment, you have complete creative freedom for this assignment. You are encouraged to work towards an aesthetically pleasing website that applies the design and development principles you have learned thus far in your academic and/or web development career. You may use Creative Commons images and/or logos with proper author attribution (provided through code comments, and/or **README.txt** or **README.md** file).

Note: Do keep in mind that as part of this assignment, you are expected to work individually, you may discuss ideas with your classmates, but do refrain from sharing any code. Though you may use Lorem Ipsum text to help you define the content hierarchy of your submission, it is recommended for you to include meaningful text where possible as it will help you see how your design may compliment the message you are looking to communicate (e.g., headings, navigation links).

- **Your lab MUST be responsive.** The level of responsiveness of your lab is dependent on the design you decide to implement. As such, you are expected to test your lab on multiple browsers, platforms, and devices.
- **Your assignment MUST be W3C compliant**, i.e., it must pass W3C front-end validations tests (e.g., HTML and CSS).

Note: Failure to submit valid code will result in a possible maximum grade of 50%. **If your assignment does not validate due to framework-specific tags or code** (e.g., browser specific CSS properties such as webkit-, framework specific HTML tags or attributes such as ng-), **these errors will be overlooked and WILL NOT affect your grade**. As well, **any validation warnings WILL NOT affect your grade**.

- **Include in your README.txt or README.md file**, the URL from which your lab can be accessed (Heroku app deployed URL). All pages you develop for this assignment will need to be accessible through that link.

Note: If you decide to use and modify any existing code, e.g., code found on online or printed sources or code used during in-class/tutorial examples, you are expected to provide author attribution in your code comments, and a more detail explanation of your sources in your README file (i.e., providing an explanation of why the piece of code is necessary for your work, where, how and why the code or section of code was modified). Keep in mind that simply stating “code was modified” does not provide sufficient information required in your programming assignments.

Submission:

- For this lab, you will need to **submit your work through Netlify, Brightspace, and GitLab as follows:**

To submit your work to Netlify:

Step 1: Creating a repository on GitHub

- Create a copy of your Gitlab repository on your personal GitHub account.
- This is just for deployment purposes ONLY, because Netlify will not allow you to deploy the app directly from Gitlab. **Code assessments will be done directly through GitLab.**

Step 2: Create an account on Netlify

- As shown on Figures 1 and 2, create a Netlify account and get it verified from your registered email address. Answer sign-up questions if required.

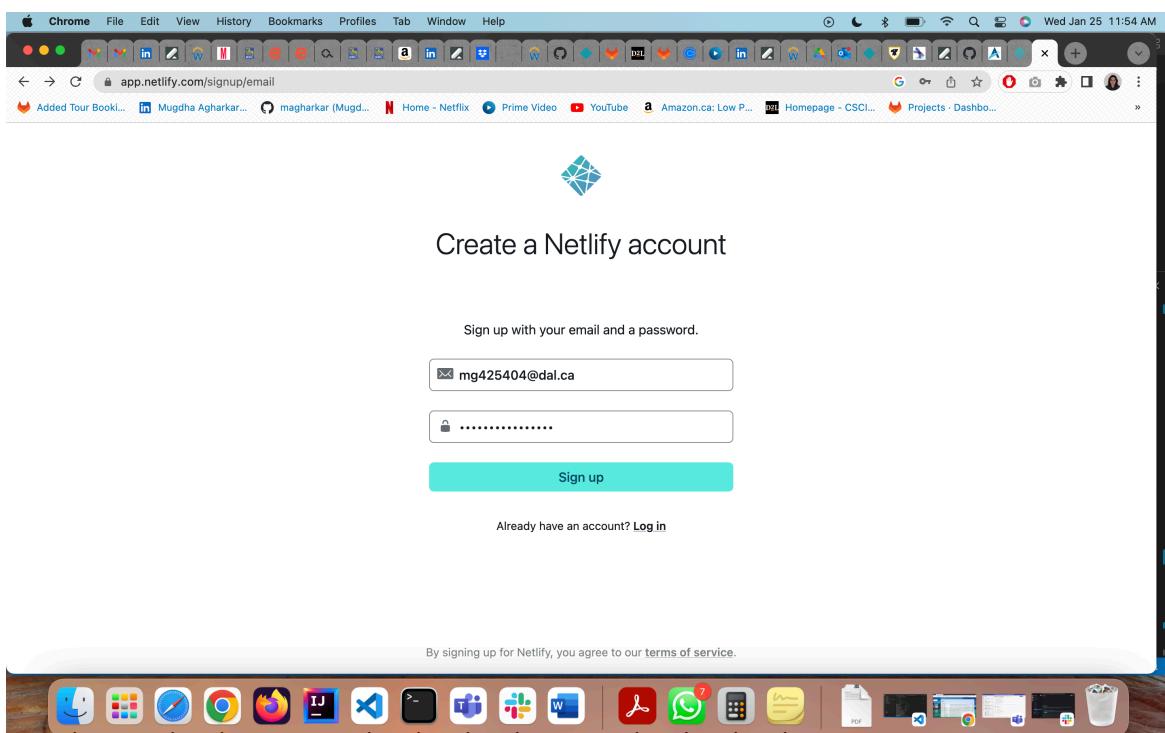


Figure 1. Creating a Netlify Account.

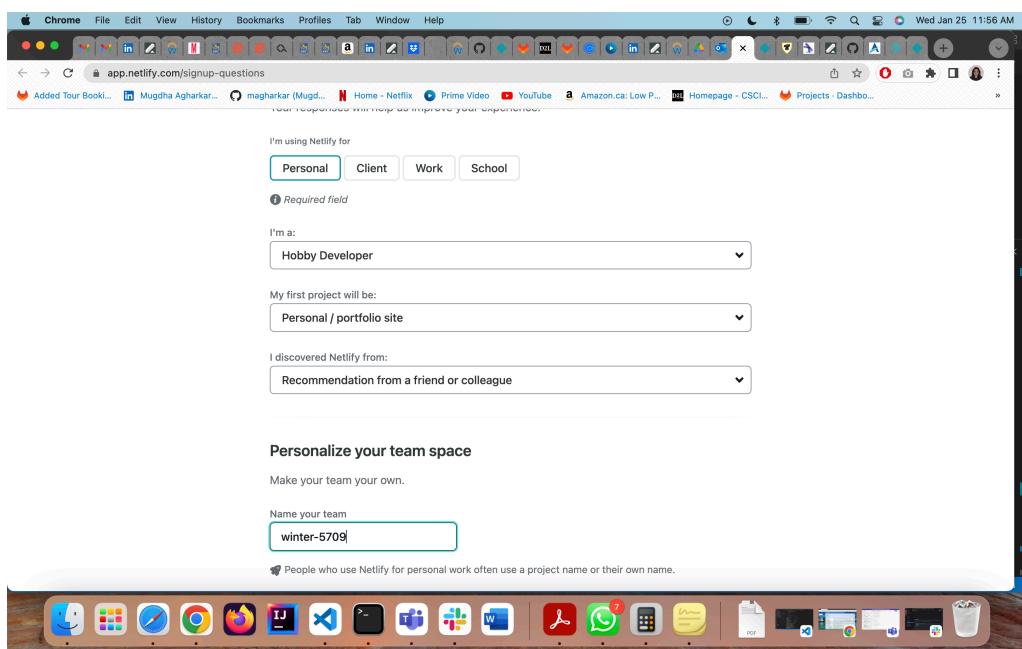


Figure 2. Creating a Netlify Account II.

Step 2: Deploying the project from GitHub

- After signing up, you will be redirected to a quick import page, where you can click on ‘import an existing project’. Note that you can also do this later in the application if you do not have a repository ready. See Figure 3.

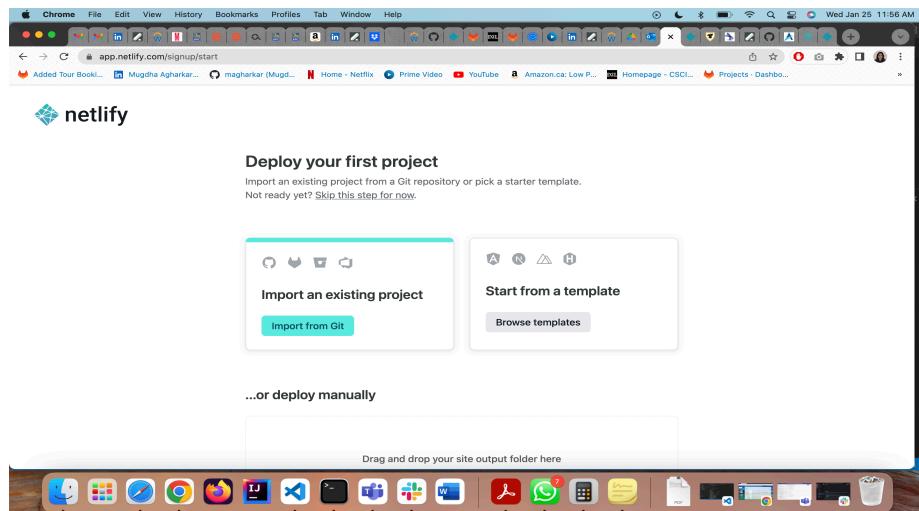


Figure 3. Deploying your first project on Netlify.

- Authorize Netlify to connect to your personal GitHub account. See Figure 4.

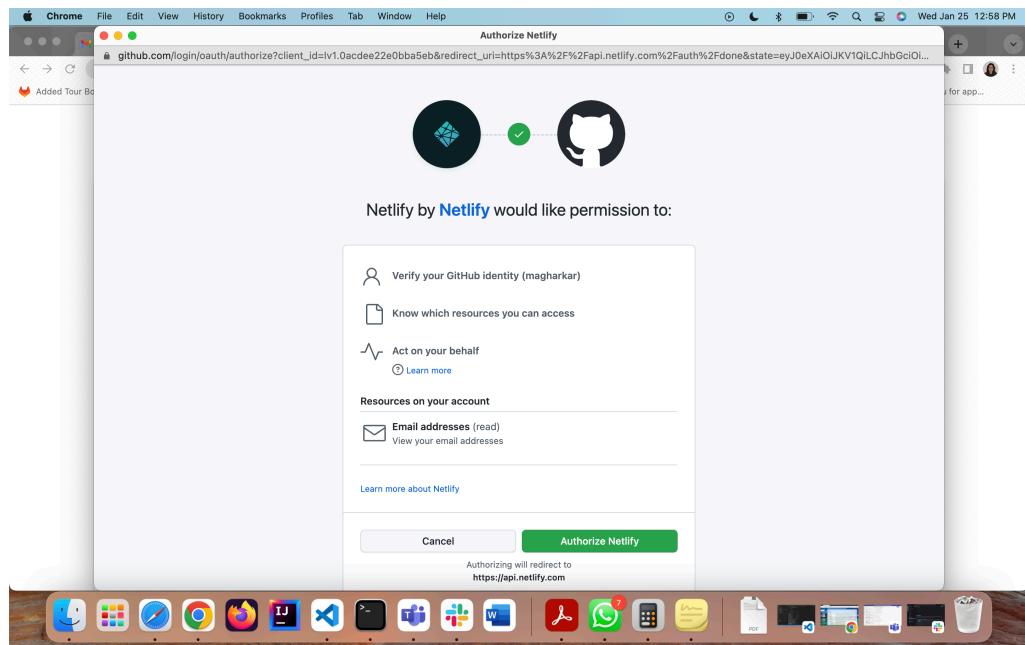


Figure 4. Connecting your GitHub account to Netlify.

- Select the branch and build options. See Figure 5.

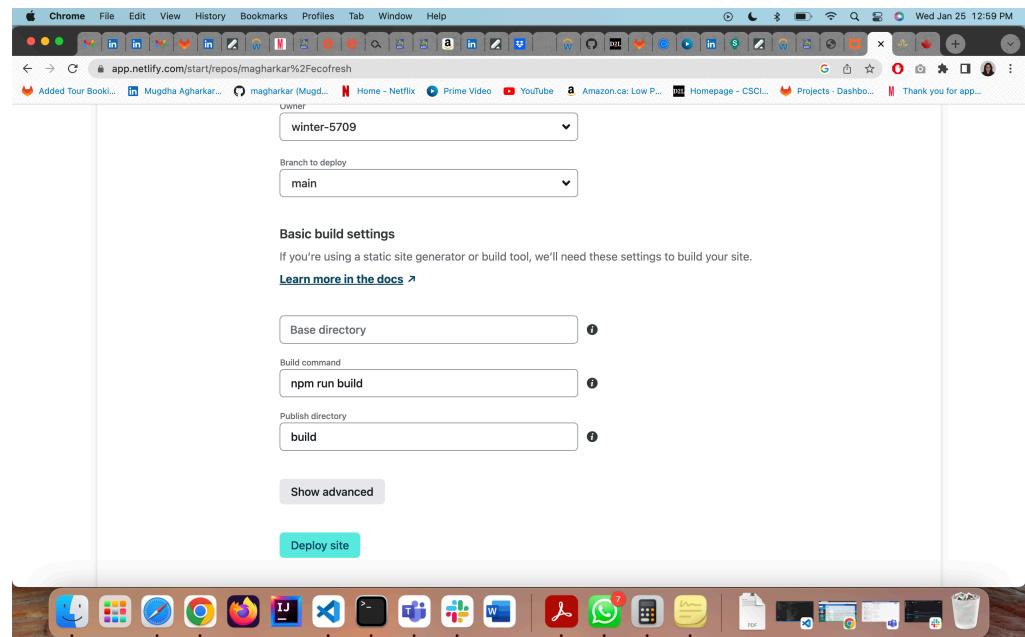


Figure 5. Selecting your branch and build options on Netlify.

- Done! Your site will be deployed in a few minutes. See Figure 6.

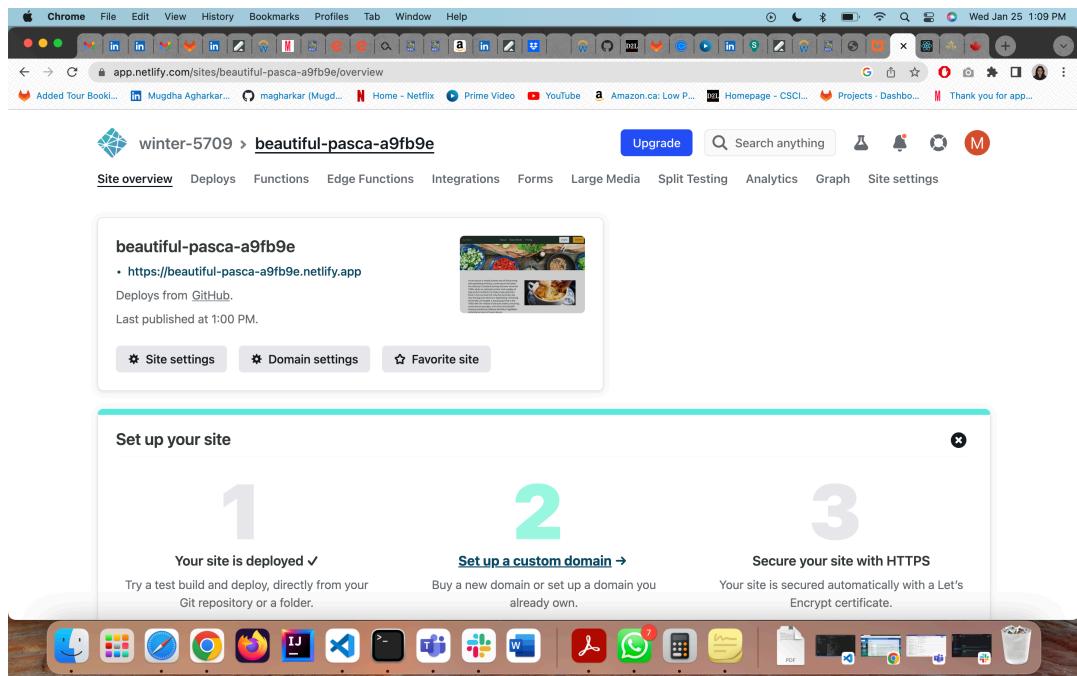


Figure 6. Completing deployment of your application on Netlify.

- Visit https://<app_name>.herokuapp.com/ on any browser and ensure you can view your work.

Note: Visit your app's URL on any browser and ensure you can view your work. Failure to submit your work through Netlify will result in a grade of **ZERO (0)**. Failure to ensure your work is remotely accessible through a web browser, using the specified URL will result in a grade of **ZERO (0)**.

- Test your lab to ensure cross-browser compatibility.
- **Include the URL to your React App in your README file**

Submitting your Work through Brightspace

- Download the **README template** available on Brightspace. *See Resources section on left-hand side menu on Brightspace.* There are TWO versions of this template, you may use whichever you feel more comfortable with.
- Edit the README template to include any citations for your code and/or images used for this Lab.

Note: If the work you are submitting as part of your Lab is work done by you without the use of any external sources, then please specify so within your README file.

- Depending on the version of the template you chose, rename your README file as:
L#_LastName_FirstName_README.md OR L#_LastName_FirstName_README.txt

Note: Ensure your README file includes the URL to your Lab for remote access.

Submitting your Work through GitLab

- Create a **git** repository on the **FCS Gitlab site**, and clone it to your local system using the following command:

```
git clone *your repo https link*
```

- Copy the HTML or JS file to the local copy of your repo and push it to the git repo using the following commands:

```
git add .
git commit -m "your commit message"
git push
```

- Setup your GITlab repo as a private project and add the course **Teaching Assistants (TAs) and Instructor as ‘Maintainers’** to your project, using their **CS IDs**. *See Lab 1 Brightspace module.*

Note: The CSIDs for this course will be provided during our lab session. Failure to add the course CS ID as 'Maintainer' for your work on GitLab will result in a maximum possible grade of 50%.

Marking Rubric:

The following grading criteria will be used for marking your lab:

| Dimensions | Does Not Meet Expectations | Meets Expectations | Exceeds Expectations |
|--|--|--|--|
| HTML (5%) | Student's HTML templates of React components are empty or not rendered properly or at all (0 - 1 points) | Student's HTML templates of React components fails to properly implement HTML tags that correctly mark-up the structure of the web document, i.e., uses the wrong tags such as <table> for defining a layout. (2 - 3 points) | Student's HTML templates of React components file successfully implements HTML tags that properly markup the structure of the web document. (5 points) |
| CSS (5%) | Student's styles.css or component CSS files are all empty or do not render properly or at all. Student did not customize any CSS frameworks used. (0 - 1 points) | Student's styles.css or component CSS files fail to properly render or are not cross-browser compatible, i.e., looks different on different browsers. Layout is not responsive. (2 - 3 points) | Student's styles.css or component CSS files successfully render and are cross-browser compatible. Lab is responsive. Student properly customized any CSS frameworks used in this lab. (5 points) |
| Design (15%) | Student's overall design is not consistent, or aesthetically pleasing. Design includes broken elements, e.g., images, links. (0 - 3 points) | Student's overall design is somewhat consistent. Includes some broken elements but it is not an overall distracting or confusing design. (5 - 9 points) | Student's overall design is consistent and aesthetically pleasing. The implemented design is overall very easy to use. (10 – 15 points) |
| React App (40%) | Student does not structure code as in React (0 - 5 points) | | Student does structure code as in React (30 - 40 points) |
| Cross-Browser Compatibility (20%) | Student's lab is not cross-browser compatible, noticeable, and distracting differences are visible. (0 points) | | Student's lab is cross-browser compatible, no noticeable or distracting differences visible. (20 points) |
| Heroku Deployment (20%) | App is not properly deployed on Heroku and no Heroku app link is provided (0 points) | | App is properly deployed on Heroku and Heroku app link is working (20 points) |
| Git repository | Code is not pushed to repo, and/or TAs and Instructor not added as maintainers. (-50 points) | | Code is properly pushed to git repo. TAs and Instructor added as maintainers. (0 points) |
| Remote Access | Student's app is not remotely accessible using expected URL. (-100 points) | | Student's app is remotely accessible using expected URL. (0 points) |

| | | |
|-------------------|--|---|
| README.txt | Student's did not submit a README.txt file and/or did not edit the template as expected. (-100 points) | Student's submitted a README.txt file properly edited to include all sources used. (0 points) |
|-------------------|--|---|