



**DALHOUSIE
UNIVERSITY**

CSCI 5308: Advance Topics in Software Development, Winter 2022

Document Type: Build Documentation

Submitted By: Group 17

Aayushi Gandhi	B00890697
Dhairya Doctor	B00864868
Shivangi Bhatt	B00863408
Saurabh Das	B00911733

Contents

Build Documentation	3
Steps to build the project on local machine:.....	3
Prerequisites:.....	3
Step 1: Installing yarn globally.....	3
Step 2: Clone the project repo	3
Step 3: Change directory to the root of the project.....	4
Step 4: Installing node_modules for frontend and backend	4
Step 5: Start backend development server	5
Step 6: Start frontend development server.....	5
Step 7: Build and serve the frontend in case you want to run the production app.....	6
Deployment in the production environment	8
Heroku environments (backend).....	8
Heroku environments (frontend)	9
Deployment environments in GitLab (for ease of access to the Heroku apps).....	10

Build Documentation

The project repo is at - <https://git.cs.dal.ca/courses/2022-winter/csci-5308/group17>

Steps to build the project on local machine:

Prerequisites:

- Node 16.14.0 (download from <https://nodejs.org/download/release/v16.14.0/>).
- Make sure Node is added to the PATH variable of the system.

```
Saurabh@SAURABH ASDC project 06:11 PM
→ node -v
v16.14.0
```

Step 1: Installing yarn globally

- Open any shell and run the command:
→ **npm i -g yarn**

Step 2: Clone the project repo

- Navigate to a suitable folder on your machine
- Clone the repo using either the HTTPS or SSH url using the command:
→ **git clone https://git.cs.dal.ca/courses/2022-winter/csci-5308/group17.git**

```
Saurabh@SAURABH ASDC project 06:11 PM
→ git clone https://git.cs.dal.ca/courses/2022-winter/csci-5308/group17.git
Cloning into 'group17'...
remote: Enumerating objects: 2223, done.
remote: Counting objects: 100% (32/32), done.
remote: Compressing objects: 100% (32/32), done.
remote: Total 2223 (delta 13), reused 0 (delta 0), pack-reused 2191
Receiving objects: 100% (2223/2223), 3.05 MiB | 3.06 MiB/s, done.
Resolving deltas: 100% (1376/1376), done.
```

- By default, the **main** branch is checked

```

Saurabh@SAURABH group17 (main) 06:37 PM
→ cd iconnect-frontend/

Saurabh@SAURABH iconnect-frontend (main) 06:37 PM
→

Saurabh@SAURABH iconnect-frontend (main) 06:37 PM
→ yarn install
yarn install v1.22.18
warning ..\..\..\package.json: No license field
[1/4] Resolving packages...
[2/4] Fetching packages...
[3/4] Linking dependencies...
warning " > react-images-uploading@3.1.3" has incorrect peer dependency "react@^16.8.0".
warning " > react-images-uploading@3.1.3" has incorrect peer dependency "react-dom@^16.8.0".
warning " > styled-components@5.3.5" has unmet peer dependency "react-is@>= 16.8.0".
[4/4] Building fresh packages...
Done in 72.38s.

Saurabh@SAURABH iconnect-frontend (main) 06:38 PM
→ cd ..

Saurabh@SAURABH group17 (main) 06:40 PM
→ cd iconnect-backend/

Saurabh@SAURABH iconnect-backend (main) 06:40 PM
→ yarn install
yarn install v1.22.18
warning ..\..\..\package.json: No license field
[1/4] Resolving packages...
[2/4] Fetching packages...
[3/4] Linking dependencies...
[4/4] Building fresh packages...
Done in 56.20s.

```

Step 3: Change directory to the root of the project

- After cloning the repo, use the command:
→ **cd group17/**

Step 4: Installing node_modules for frontend and backend

- Change directory to iconnect-frontend
Run the command → **yarn install**
- Change back to the parent directory and go to iconnect-backend
Run the command → **yarn install**

This installs all the required node_modules (dependencies) for running the project.

Step 5: Start backend development server

- From the root directory, change directory to iconnect-backend/ and run the command:
→ **yarn start**

```
Saurabh@SAURABH iconnect-backend (main) 06:41 PM
→ yarn start
yarn run v1.22.18
warning ..\..\..\..\package.json: No license field
$ node src/index.js
body-parser deprecated undefined extended: provide extended option src\index.js:31:17
Running on port 3200
Successfully connected to MongoDB!
```

- This starts the backend node server on the local 3200 port

Step 6: Start frontend development server

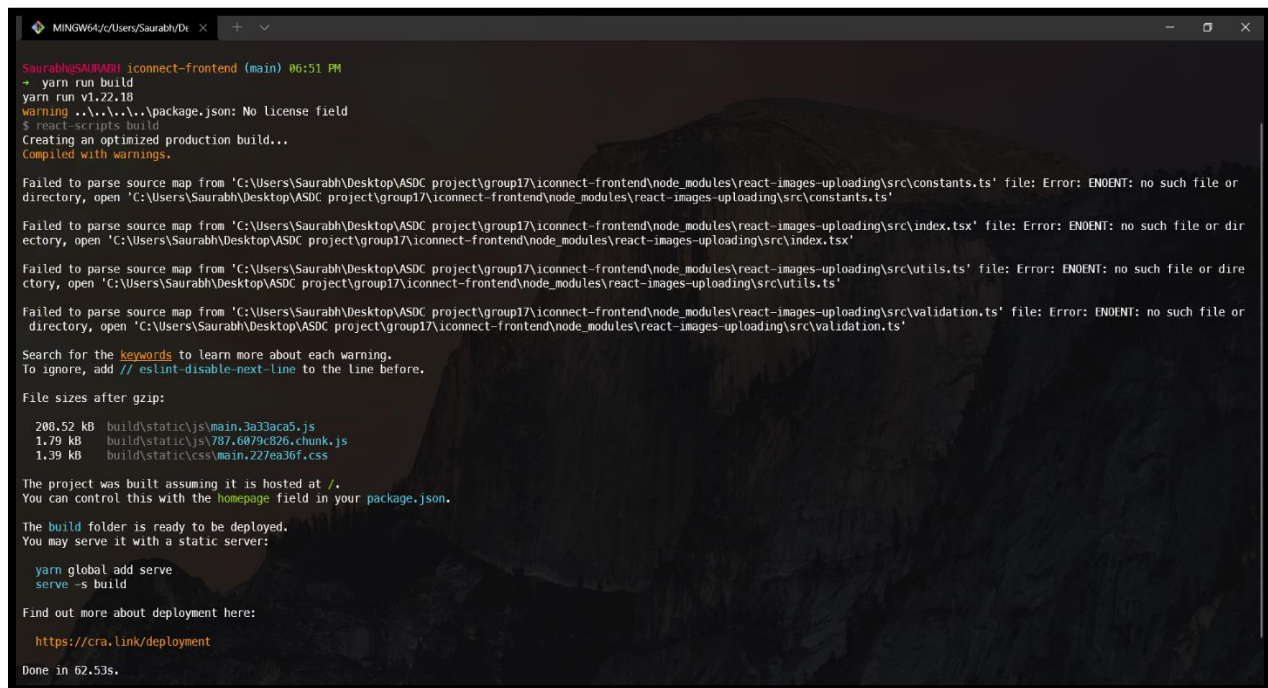
- From the root directory, change directory to iconnect-frontend/ and run the command:
→ **yarn start**

```
Saurabh@SAURABH iconnect-frontend (main) 06:46 PM
→ yarn start
yarn run v1.22.18
warning ..\..\..\..\package.json: No license field
$ react-scripts start
(node:14032) [DEP_WEBPACK_DEV_SERVER_ON_AFTER_SETUP_MIDDLEWARE] DeprecationWarning: 'onAfterSetupMiddleware' option is deprecated. Please use the 'setupMiddlewares' option.
(Use `node --trace-deprecation ...` to show where the warning was created)
(node:14032) [DEP_WEBPACK_DEV_SERVER_ON_BEFORE_SETUP_MIDDLEWARE] DeprecationWarning: 'onBeforeSetupMiddleware' option is deprecated. Please use the 'setupMiddlewares' option.
Starting the development server...
```

- This starts the frontend react app on port 3000
- It will open your default browser with the homepage up

Step 7: Build and serve the frontend in case you want to run the production app

- In case you want to product an optimized build to serve
- Install the “serve” node package using the command “**npm i -g serve**”
- Then change directory to iconnect-frontend/ and run the command “**yarn run build**”



```
MINGW64/c/Users/Saurabh/D... x + -
Saurabh@SAURABH iconnect-frontend (main) 06:51 PM
→ yarn run build
yarn run v1.22.18
warning ..\..\..\package.json: No license field
$ react-scripts build
Creating an optimized production build...
Compiled with warnings.

Failed to parse source map from 'C:\Users\Saurabh\Desktop\ASDC project\group17\iconnect-frontend\node_modules\react-images-uploading\src\constants.ts' file: Error: ENOENT: no such file or directory, open 'C:\Users\Saurabh\Desktop\ASDC project\group17\iconnect-frontend\node_modules\react-images-uploading\src\constants.ts'

Failed to parse source map from 'C:\Users\Saurabh\Desktop\ASDC project\group17\iconnect-frontend\node_modules\react-images-uploading\src\index.tsx' file: Error: ENOENT: no such file or directory, open 'C:\Users\Saurabh\Desktop\ASDC project\group17\iconnect-frontend\node_modules\react-images-uploading\src\index.tsx'

Failed to parse source map from 'C:\Users\Saurabh\Desktop\ASDC project\group17\iconnect-frontend\node_modules\react-images-uploading\src\utils.ts' file: Error: ENOENT: no such file or directory, open 'C:\Users\Saurabh\Desktop\ASDC project\group17\iconnect-frontend\node_modules\react-images-uploading\src\utils.ts'

Failed to parse source map from 'C:\Users\Saurabh\Desktop\ASDC project\group17\iconnect-frontend\node_modules\react-images-uploading\src\validation.ts' file: Error: ENOENT: no such file or directory, open 'C:\Users\Saurabh\Desktop\ASDC project\group17\iconnect-frontend\node_modules\react-images-uploading\src\validation.ts'

Search for the keywords to learn more about each warning.
To ignore, add // eslint-disable-next-line to the line before.

File sizes after gzip:

 208.52 kB build\static\js\main.3a30aca5.js
  1.79 kB build\static\js\787.6079c826.chunk.js
  1.39 kB build\static\css\main.227ea36f.css

The project was built assuming it is hosted at /.
You can control this with the homepage field in your package.json.

The build folder is ready to be deployed.
You may serve it with a static server:

  yarn global add serve
  serve -s build

Find out more about deployment here:

  https://cra.link/deployment

Done in 62.53s.
```

- This creates a production ready build of the app which can be served from the local machine using the command:
➔ “**serve -s build**”

```
Saurabh@SAURABH iconnect-frontend (main) 06:55 PM  
→ serve -s build
```

Serving!

- Local: <http://localhost:3000>
- On Your Network: <http://192.168.4.73:3000>

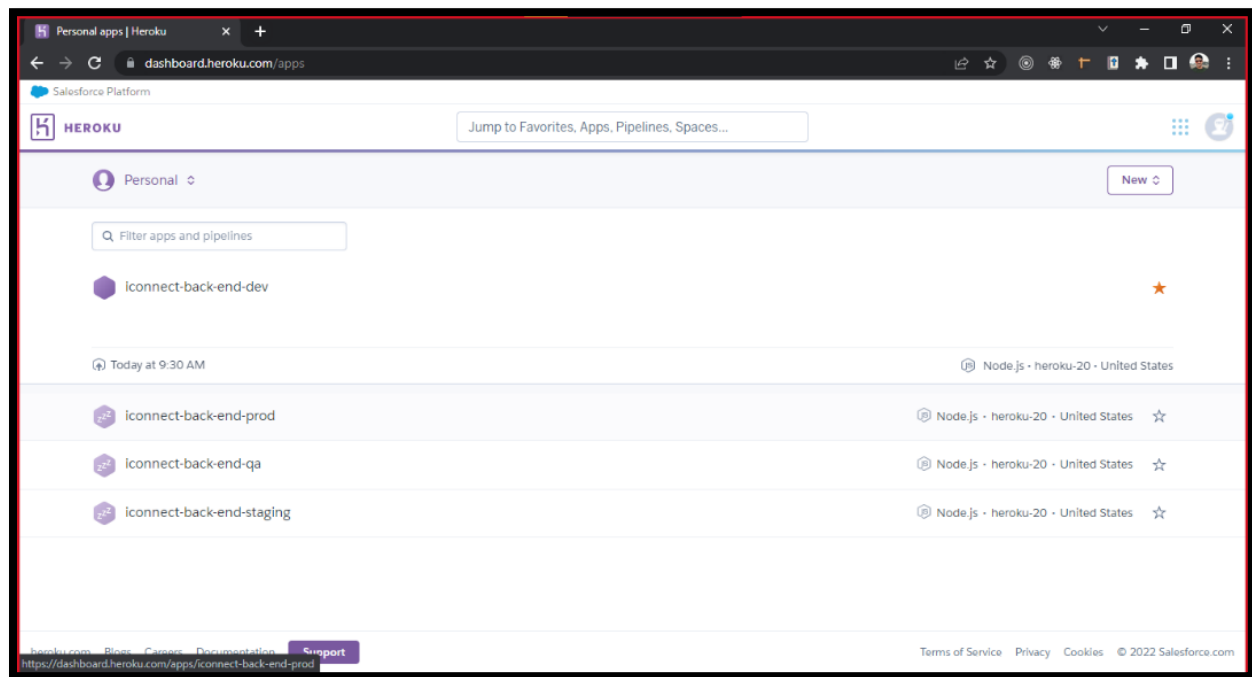
copied local address to clipboard

- The production build uses the backend server

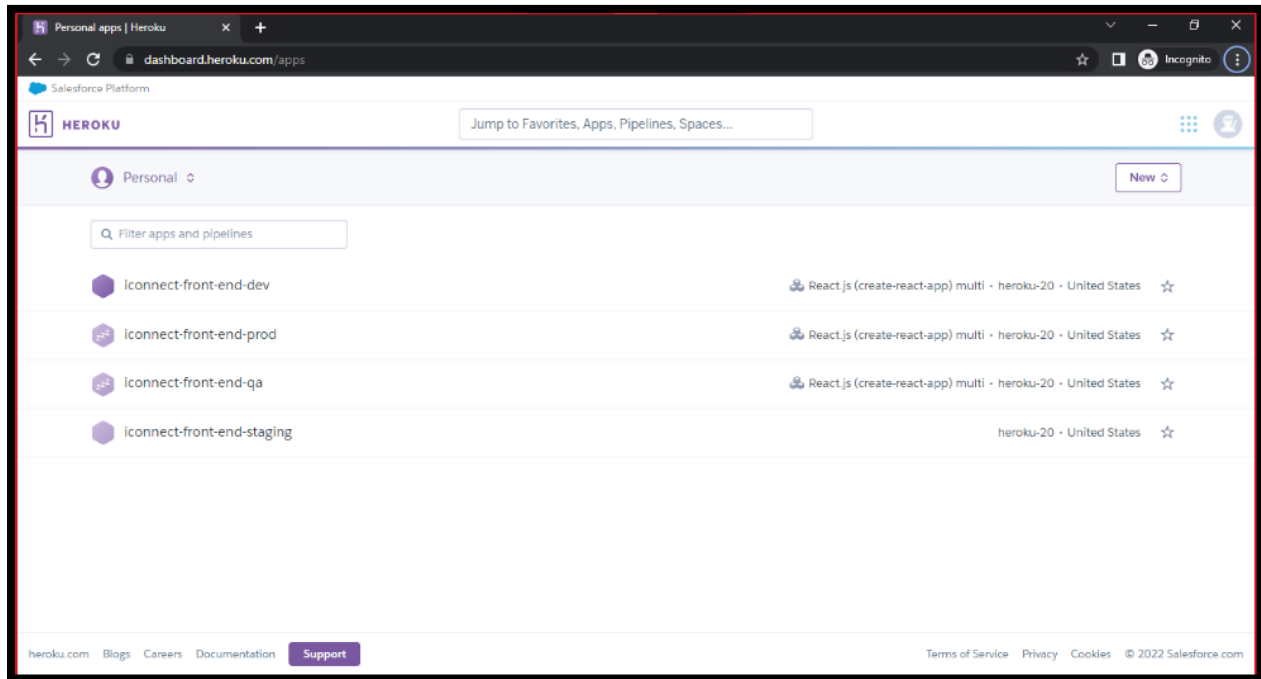
Deployment in the production environment

- There are 8 production environments set (4 for backend and 4 for frontend)
- Backend and frontend both have their own “dev”, “qa”, “stage”, and “prod” environment.
- The deployment environments match the branch names in our repo. (main branch deploys to prod env)

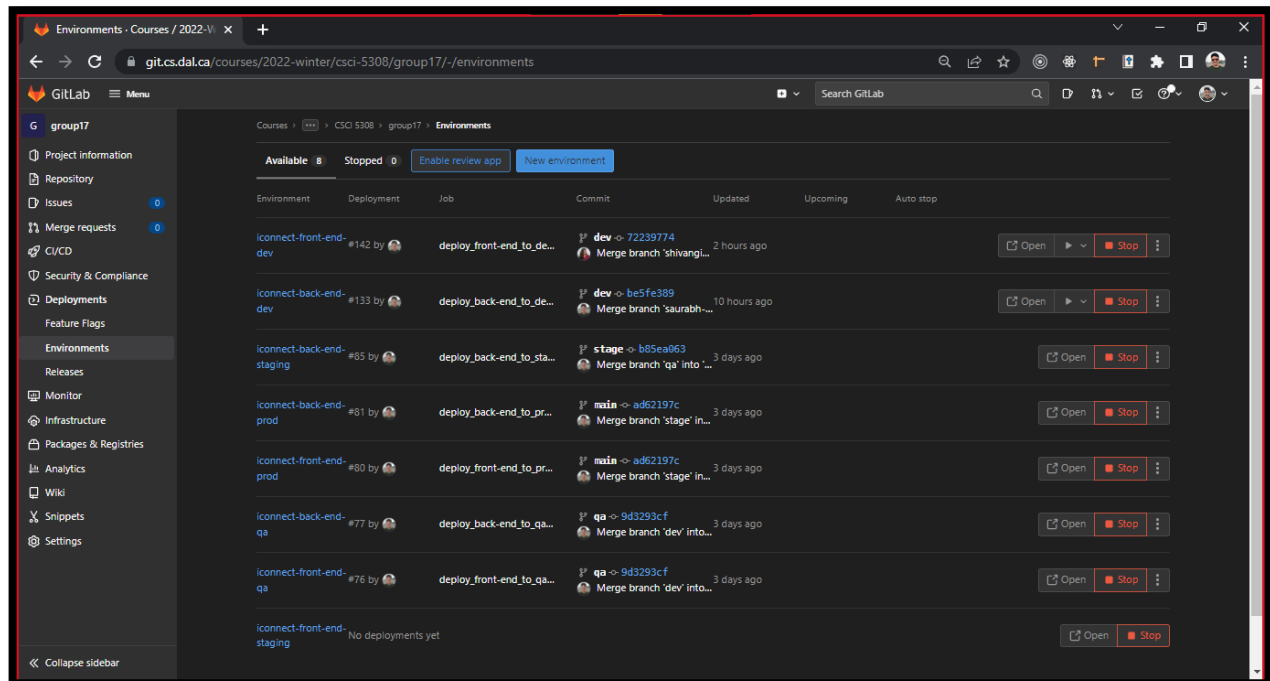
Heroku environments (backend)



Heroku environments (frontend)



Deployment environments in GitLab (for ease of access to the Heroku apps)



Final Production URL:

<https://iconnect-front-end-prod.herokuapp.com/login>