**Hardhat Enterprises**

**Fortify Project (Front End) Handover**

REVISION HISTORY

|  |  |  |
| --- | --- | --- |
| **Date** | **Summary of Changes** | **Updated By** |
| 23-May-2022 | Baseline handover document with details and information about on the front-end project. | Rex Ambrosio  (217054465) |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

TABLE OF CONTENTS

[1. Purpose of this document 4](#_Toc104172299)

[2. Overview 4](#_Toc104172300)

[3. The Frontend Project 4](#_Toc104172301)

[3.1 Frontend project (React) 4](#_Toc104172302)

[3.2 Project structure 4](#_Toc104172303)

[4. Working with Git 5](#_Toc104172304)

[4.1 Getting started with Git 5](#_Toc104172305)

[3.3.1 Install Git on Mac 5](#_Toc104172306)

[3.3.2 Install Git on Windows 5](#_Toc104172307)

[3.3.3 Install Git on Linux (Debian/Ubuntu) 5](#_Toc104172308)

[3.3.4 Install Git on Linux (RHEL/Fedora) 5](#_Toc104172309)

[4.2 Configuring Git username and email 6](#_Toc104172310)

[4.3 Working collaboratively with Git 6](#_Toc104172311)

[3.5.1 Cloning the project repository 6](#_Toc104172312)

[3.5.2 Collaborative workflow using git pull, git add, git commit and git push 6](#_Toc104172313)

[3.5.3 Branching and merging 7](#_Toc104172314)

[5. Helpfortify.org domain and Domain Registrar 7](#_Toc104172315)

[5.1 GoDaddy Login Credentials 7](#_Toc104172316)

[5.2 Managing DNS 8](#_Toc104172317)

[5.3 Creating CNAME record for SSL certificate’s domain validation 8](#_Toc104172318)

[6. SSL Certificate 8](#_Toc104172319)

[6.1 Requesting CA signed SSL certificate from Zero SSL 9](#_Toc104172320)

[7. Resources 10](#_Toc104172321)

# Purpose of this document

This document is prepared to give some guidance and a head start for the team who will be taking over the project during the next Capstone program. The project details, repository location, account login credentials, tools and products used in procuring some of the deliverables of the project are included in this document. Some basic instructions and guidance are included here including steps in reproducing but more information is available on the resources provided for each section which we recommended to read further.

# Overview

The Hardhat Enterprises Fortify project is a website platform that connects security cohorts and organisations together to provide help and even information about security. It’s been decided that the front-end will run on React as its technology stack and build from there due to numerous benefits it can offer. A domain name has been registered named ‘helpfortify.org’ and an SSL certificate has been procured as well as a prepared for when the project is decided to be rolled out into production or even developed into a proof-of-concept functional website.

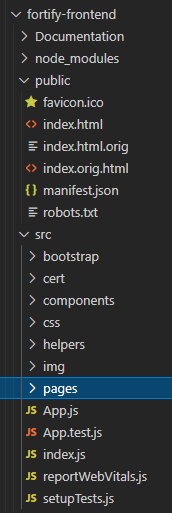
# The Frontend Project

## Frontend project (React)

The frontend project is located at the Git repository specified below. The contents of the project can be copied over for the next capstone program and a new repository may be created if in such case that the owner of the repository is no longer available.

<https://bitbucket-students.deakin.edu.au/scm/~rambrosio/hh-fortify-webdev.git>

## Project structure



Below is the current high-level structure of the project. The structure of the React project and

* **public** – contains the public web documents including the default html document.
* **src** – contains React application objects that needs to be reference internally within the React application.
* **src/bootstrap** – contains the bootstrap javascrip for additional stylish componnents
* **src/cert** – the SSL certificate, private key and the chain certificate for the website
* **src/components** – contains some of the function components including Navbar, Footer and the Keycloak authentication.
* **src/css** – contains the external cascading stylesheets used for the website
* **src/helpers** – some additional function for authentication
* **src/img** – contains the images used for the website
* **src/pages** – contains the individual page components of the website

# Working with Git

## Getting started with Git

To get started, make sure that ‘git’ package is installed on your machine. If you have not done so, you could follow some of the steps below or refer to <https://www.atlassian.com/git/tutorials/install-git> for a more detailed instruction.

### Install Git on Mac

* Download the latest [Git for Mac installer](https://sourceforge.net/projects/git-osx-installer/files/).
* Follow the prompts to install Git.
* Open a terminal and verify the installation was successful by typing git --version:

### Install Git on Windows

* Download the latest [Git for Windows installer](https://git-for-windows.github.io/).
* When you've successfully started the installer, you should see the **Git Setup** wizard screen. Follow the **Next** and **Finish** prompts to complete the installation. The default options are pretty sensible for most users.
* Open a Command Prompt (or Git Bash if during installation you elected not to use Git from the Windows Command Prompt).

### Install Git on Linux (Debian/Ubuntu)

* Open a terminal and install git using apt-get:
  + **$ sudo apt-get update**
  + **$ sudo apt-get install git**
* Verify the installation was successful by typing git --version:
  + **$ git --version**
  + **git version 2.9.2**

### Install Git on Linux (RHEL/Fedora)

* From your terminal, install Git using dnf (or yum, on older versions of Fedora):
  + **$ sudo dnf install git**
  + **$ sudo yum install git**
* Verify the installation was successful by typing git --version:
  + **$ git --version**
  + **git version 2.9.2**

## Configuring Git username and email

Once git is installed on your local machine, the username and email has to be configured to associate your changes and commits done on your local environment.

* From your terminal, configure your git environment username and email by entering the following commands:
  + **$ git config --global user.name "Your Name"**
  + **$ git config --global user.email "youremail@deakin.edu.au"**

## Working collaboratively with Git

There are different ways to work collaboratively as a team using Git. In Deakin’s Atlassian Bitbucket, collaborators are specified according to repository permissions and branching permissions. For simplicity, branching permissions are not enforced in this guide so the team will have the flexibility to commit, merge and push their updates into the repository.

### Cloning the project repository

Once the git username and email is set, users can then clone the repo by running the git clone command below:

**git clone <bitbucketURL>/git/<projectname>/<reponame>.git**

Specifically for this project, you can run the command to clone the Fortify frontend project:

**git clone https://bitbucket-students.deakin.edu.au/scm/~rambrosio/hh-fortify-webdev.git**

### Collaborative workflow using git pull, git add, git commit and git push

As a basic collaborative workflow, it is important to make sure that you have the updated version of the repository on your local computer. This is done by running ‘git pull’ before adding and committing a change. A sample workflow is done by following the steps below <https://pages.nist.gov/git-novice-MSE/08-collab/index.html>:

* Update your local repo with
  + **git pull origin master**
* Make your changes and stage them with
  + **git add <file>**
* Commit your changes with
  + **git commit -m “your comment/remarks”**, and then
* Upload the changes to the Git repository with
  + **git push origin master**

### Branching and merging

To avoid accidental changes of the whole project during collaboration, it is always better to create branches and testing those changes on that branch instead of making changes from the main branch. The workflow below outlines how to branch out and test changes before merging it to the main branch and pushing the change to the remote repository <https://www.atlassian.com/git/tutorials/using-branches/git-merge>:

* Start a new branch
  + **git checkout -n newbranchname**
* edit some files, test the changes and then commit
  + **git add <file>**
  + **git commit -m “your comment/remarks”**
* Go back to the main branch
  + **git checkout main**
* Ensure that you have the latest copy of the remote repository
  + **git pull origin master**
* Merge the sub-branch to the main branch
  + **git merge newbranchname**
* Upload the changes to the Git repository with
  + **git push origin master**
* Delete the branch created
  + **git branch -d newbranchname**

# Helpfortify.org domain and Domain Registrar

There’s a domain name for the Foritfy project’s frontend that is ‘**helpfortify.org**’ which has been registered thru GoDaddy (<https://www.godaddy.com>). The following sections provides the login credentials and working with DNS for resolutions and SSL certificate domain validation.

## GoDaddy Login Credentials

To login to the domain registrar, go GoDaddy ([https://www.godaddy.com](https://www.godaddy.com/en-sg)) and on the top right section of the page, select the ‘Sign In’ button. You can either login using the username or customer number below together with the password.

Username: **fortifolks**

Customer #: **469267353**

Password: **ZAQ!2wsxCDE#4rfv**

For the purpose of the handover, the 2-step authentication has been removed but you can set this up to secure the account ‘Account Settings’.

## Managing DNS

Once logged in on GoDaddy, you can access the domain’s free basic DNS management under ‘My Products’ and following some guided process below.

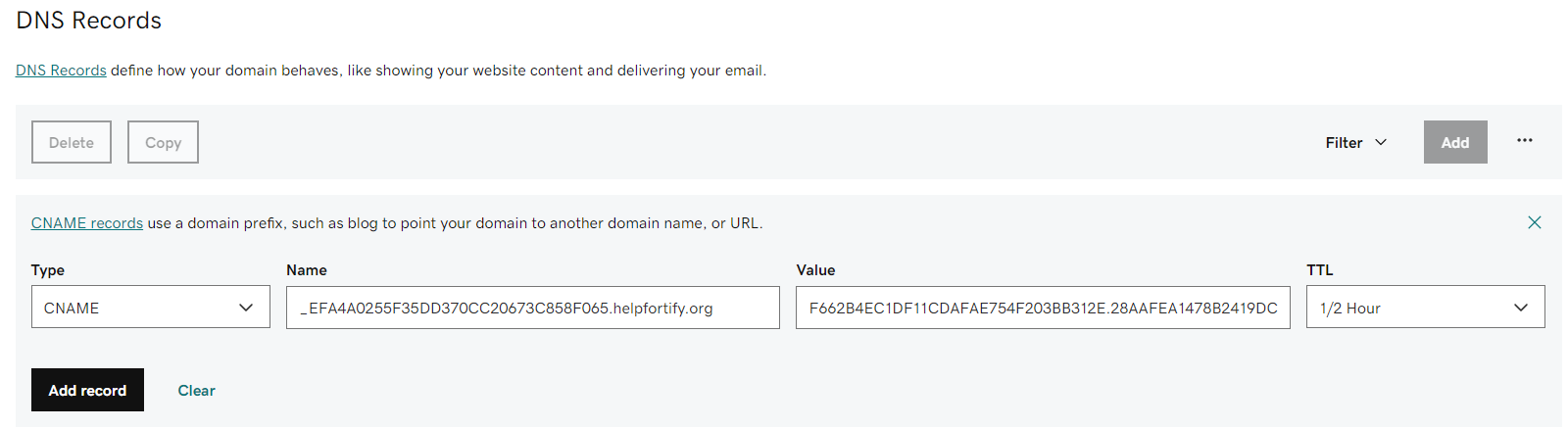
* Under ‘My Products, select ‘Manage my domain’ under ‘helpfortify.org’ domain.
* On the left section of the page, select ‘Domain’
* Under helpfortify.org section on the right, select ‘Manage DNS’
* There’s an initial A record pointed to an IP but this can be changed accordingly when the website is hosted somewhere in the future.

Managing DNS would be required especially when the domain hosted and rolled out to production. At the same time, it may also be necessary when an SSL certificate is requested and the Certification Authority requires some domain verification which can also be done using DNS CNAME or TXT records. More information can on managing DNS from the registrar on the this link <https://sg.godaddy.com/help/manage-dns-records-680>.

## Creating CNAME record for SSL certificate’s domain validation

As you are logged into GoDaddy’s DNS management, you can create different DNS records while using GoDaddy’s affiliated DNS hosting service providers. For SSL certificate domain validation, this will require you to create a CNAME record and you may follow the steps below to get you through the validation.

* Under ‘DNS Management’, click ‘Add’ and enter the details as instructed by the SSL certificate provider.



* Select ‘Add Record’ and the CNAME record will be created which will be verified during the SSL certificate domain validation.

# SSL Certificate

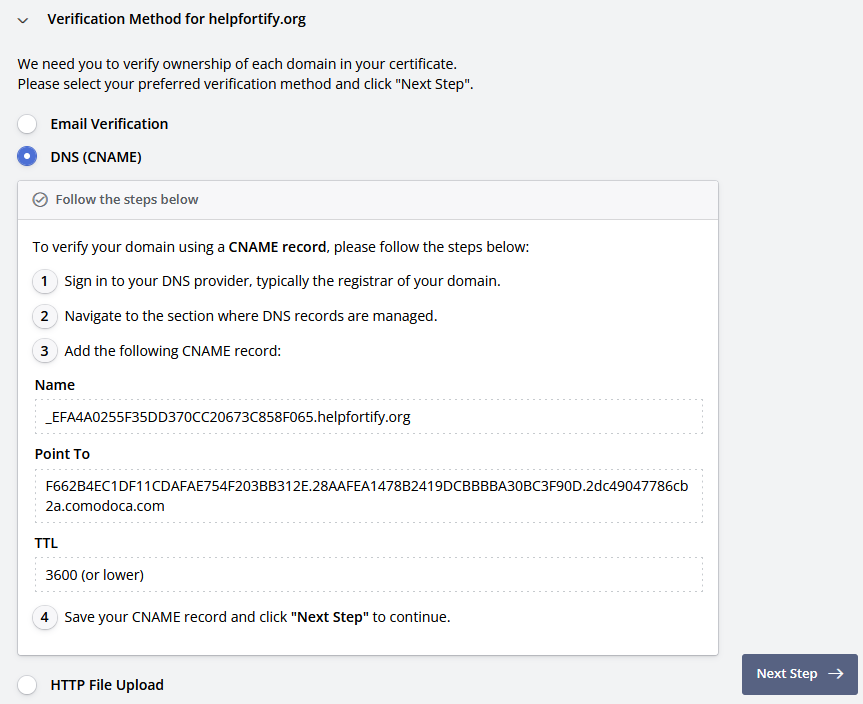
Additionally, there’s an SSL certificate prepared for the domain’s website acquired thru ‘Zero SSL’ (<https://zerossl.com/>). It is free but is only valid for 90 days. You may follow the steps below in requesting for a new SSL certificate if it expires.

## Requesting CA signed SSL certificate from Zero SSL

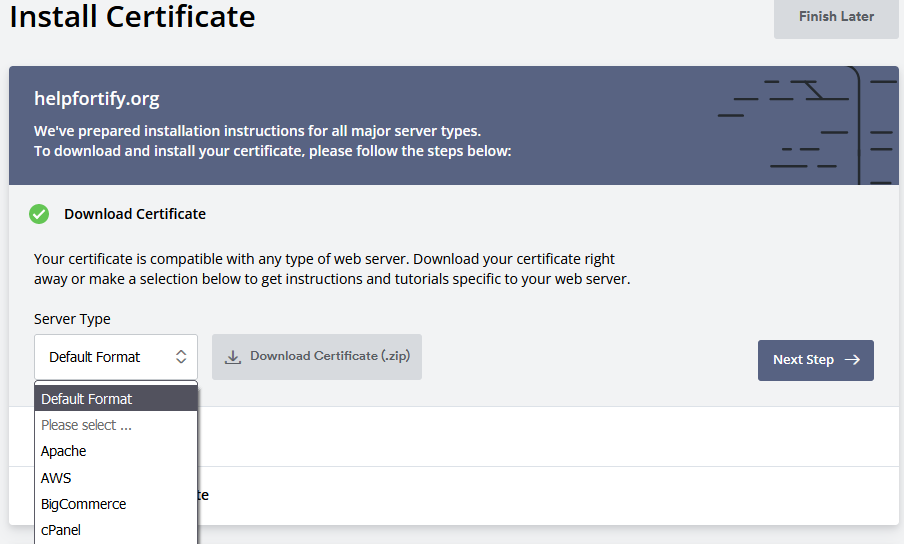
Go to <https://zerossl.com/> and enter the domain ‘helpfortify.org’ under ‘Create Free SSL Certificate’ form and then proceed to the next step. You will be redirected to a page where you need to register an account to be able to proceed. Start registering an account or log in when asked.

* Once logged in, select ‘New Certificate’ button to create a new SSL certificate
* Enter the domain ‘helpfortify.org’ under ‘Enter Domains’ form and proceed to the next step. You can also specify if a wildcard certificate is needed in this step in case necessary.
* Under ‘Validty’ step, select ’90-Day Certificate’ and proceed to the next step.
* Under ‘CSR and Contact’, unselect ‘Auto-Generate CSR’ and put your contact information accordingly into the provided fields. This information will be included in the certificate details once generated. Alternatively, if you have generated your own CSR (certificate signing request), you can paste the base64 encoded data into the form before proceeding to the next step.
* Under ‘Finalize Your Order’, select ‘Free’ and then proceed to the next step.

These steps will process the creation of your requested SSL certificate but will require some verification to ensure that you do own the domain. You will be redirected to a page that provides some options for verification and in this case, choose ‘DNS (CNAME)’ as your method of verification. A sample instruction will be provided on what you need to configure on your DNS in order to verify the domain successfully.



* Follow the instructions and enter the Name and Point To record under your DNS CNAME record in GoDaddy and once that is done, you can proceed to the next step.
* Once the DNS CNAME record is done, you can start the domain verification by selecting ‘Verify Domain’. The domain verification process will start and complete and you will have the options to download your CA signed SSL certificate according to platform/technology you’re using as shown in the image below.



* Download the certificate and store it in a safe place. The archive download will contain the signed SSL certificate, the private key and the chain certificates that you will use to deploy in your environment.

# Resources

<https://www.w3schools.com/react/default.asp>

<https://www.atlassian.com/git/tutorials/learn-git-with-bitbucket-cloud>

<https://confluence.atlassian.com/bitbucketserver0710/getting-started-with-git-and-bitbucket-server-1044104766.html>

<https://pages.nist.gov/git-novice-MSE/08-collab/index.html>

<https://www.atlassian.com/git/tutorials/using-branches/git-merge>

[https://www.godaddy.com](https://www.godaddy.com/en-sg)

<https://zerossl.com/>