**CS 628 Full-Stack Development II**

**HOS03A Creating an Express Project**

July 16, 2020

12/29/2020 Updated by Min

School of Technology & Computing (STC)

City University of Seattle (CityU)

**Before You Start**

* The directory path shown in screenshots may be different from yours.
* Some steps are not explained in the tutorial**.** If you are not sure what to do:
  1. Consult the resources listed below.
  2. If you cannot solve the problem after a few tries, ask a TA for help.

**Resources**

* Holmes, S. & Harber, C. (2019). [Getting MEAN with Mongo, Express, Angular, and Node, Second Edition](https://login.proxy.cityu.edu/sso/skillport?context=148121). Manning Publications. (ISBN 9781617294754)
  + Chapter 3: Creating and Setting up a MEAN Project
    - 3.2 Creating an Express Project
  + Appendix A Installing the Stack- Installing Express Globally
  + Appendix B: Installing and Preparing the Supporting Cast - Setting up Heroku

**Learning Outcomes**

* Understand how to create scaffolds for an Express project.
* Understand how to create a Platform as a Service (PaaS) environment.
* Create an Express project and deploy it to a cloud PaaS.

**Activities**

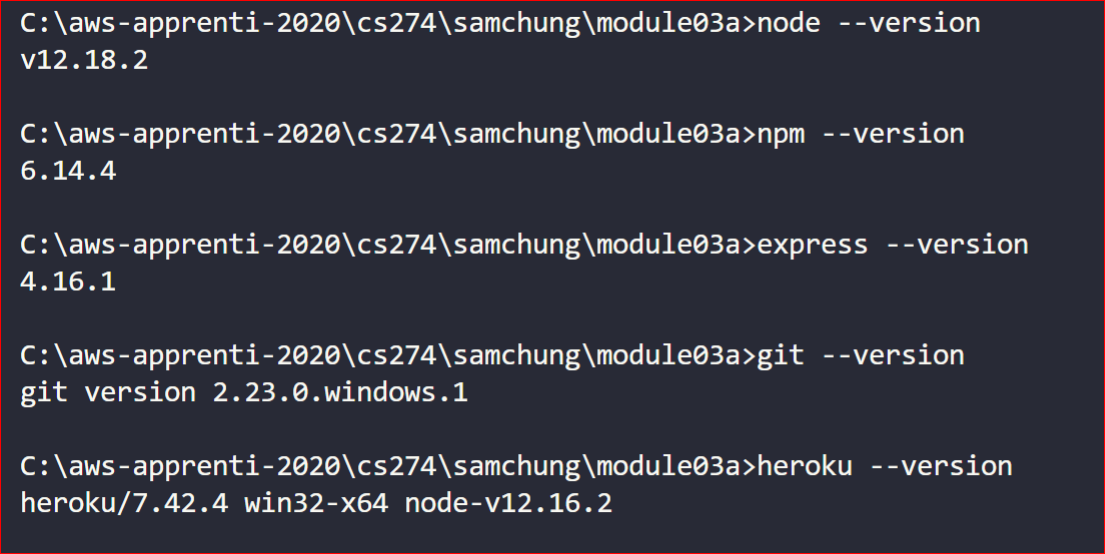
* Preparing for a MEAN project environment
* **Verifying the Installations**
* **Creating an Express Project with Scaffolds**
* **Deploying Your Project onto a Cloud PaaS**
* Source Code Management and Deployment

**Preparing for a MEAN project environment**

1. Create the “module03a” folder under repo:  
   >>>mkdir module03a  
   
2. Open the “module03a” folder in the VSCode.  
   Close the “Welcome” window.  
   Open the terminal (Ctrl+`).-

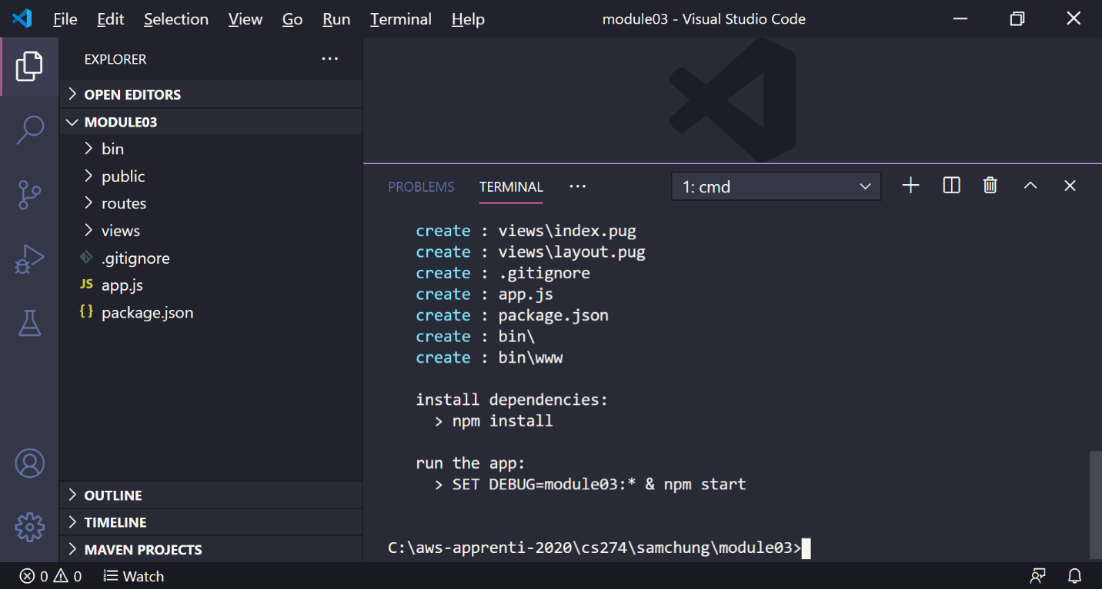
**Verifying the Installations**

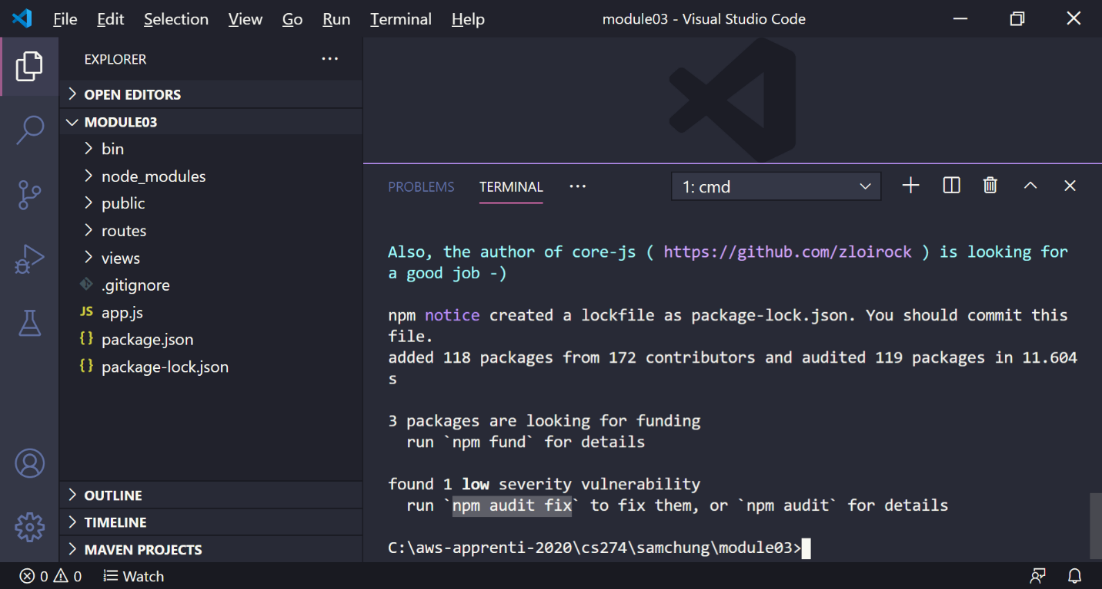
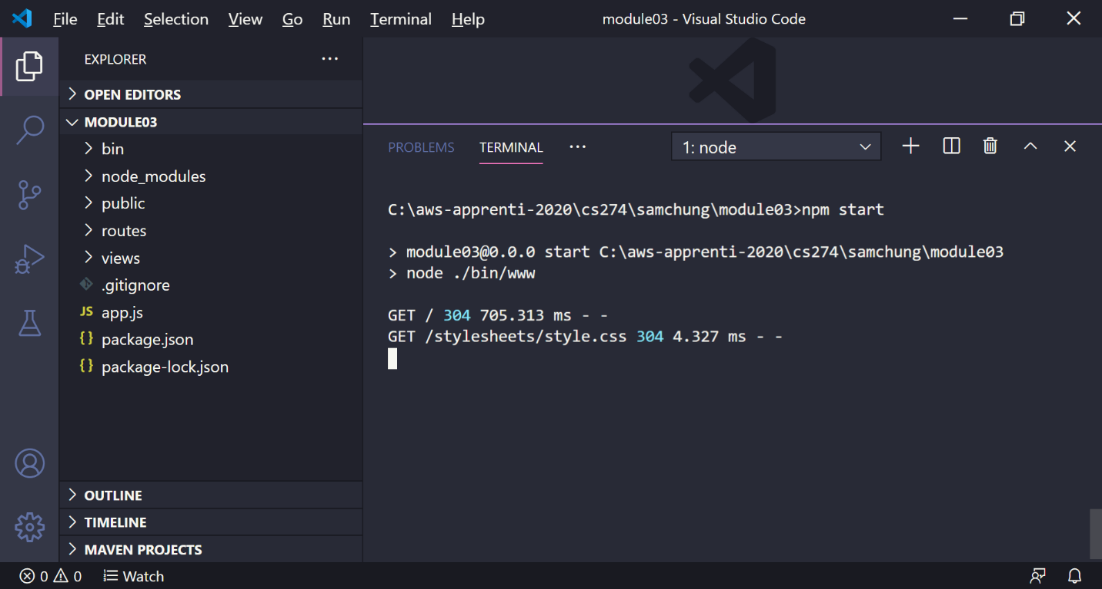
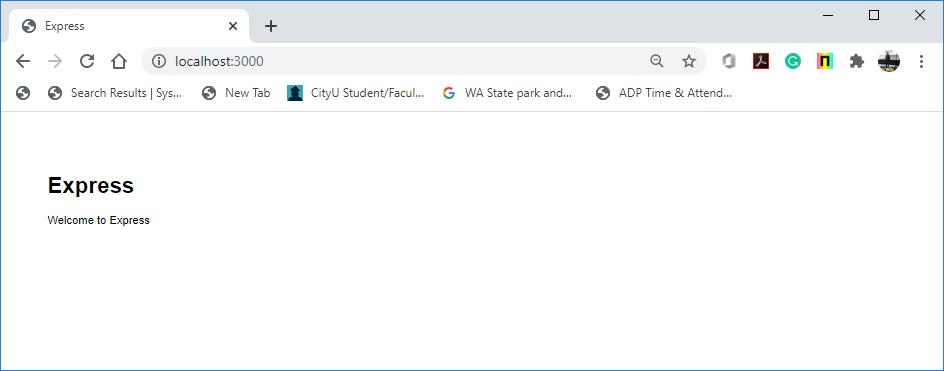
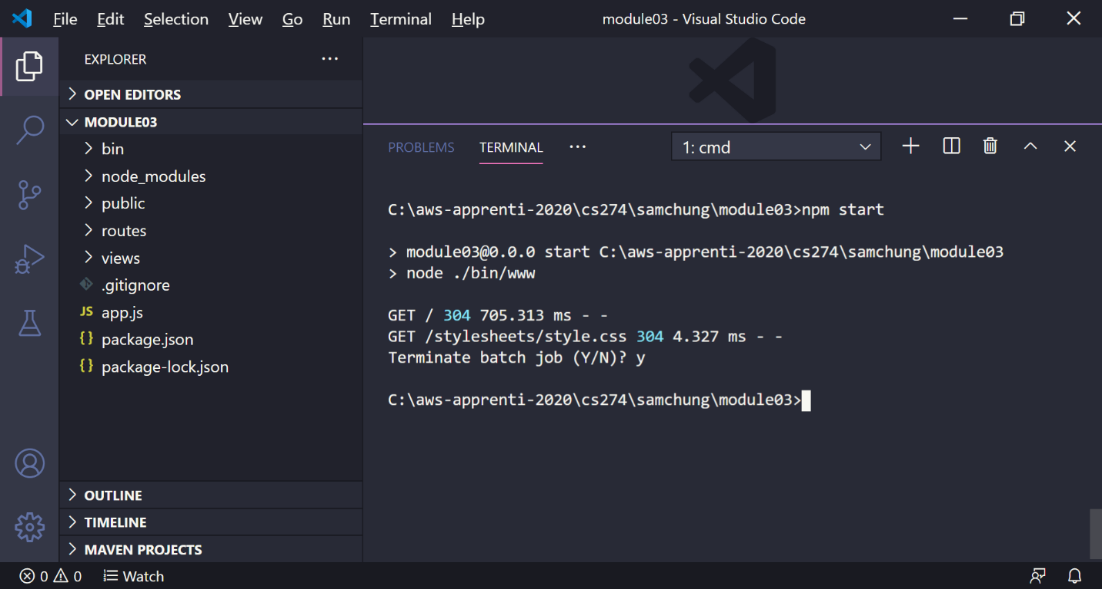
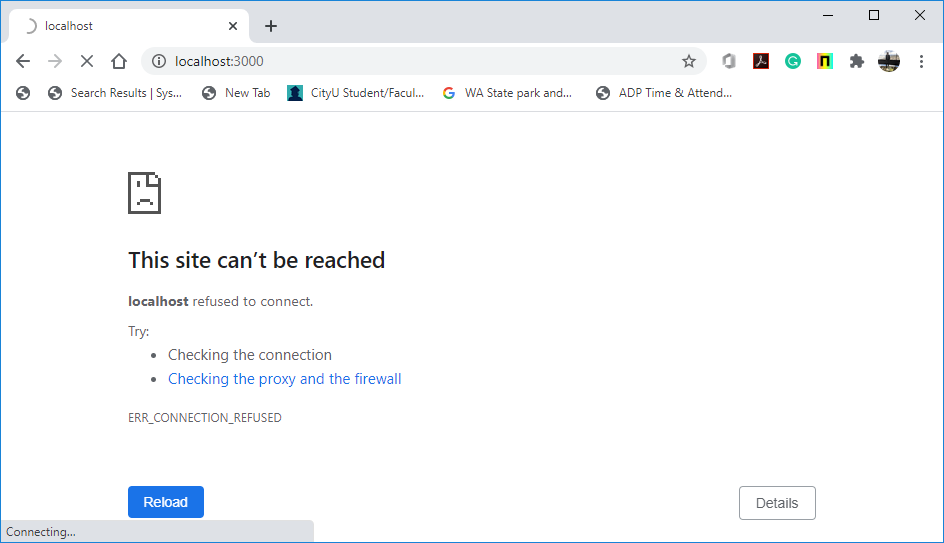
1. Check whether you have the following five things:
   1. A JavaScript runtime environment  - node  
      >>>node --version
   2. a package manager for the JavaScript programming language – npm  
      >>>npm --version
   3. A server-side web application framework – express  
      >>>express --version
   4. A distributed version control system – git  
      >>>git --version
   5. A cloud platform as a service (PaaS) (**If you did not install it, you will do it in this HOS later**.   
      >>>heroku –version



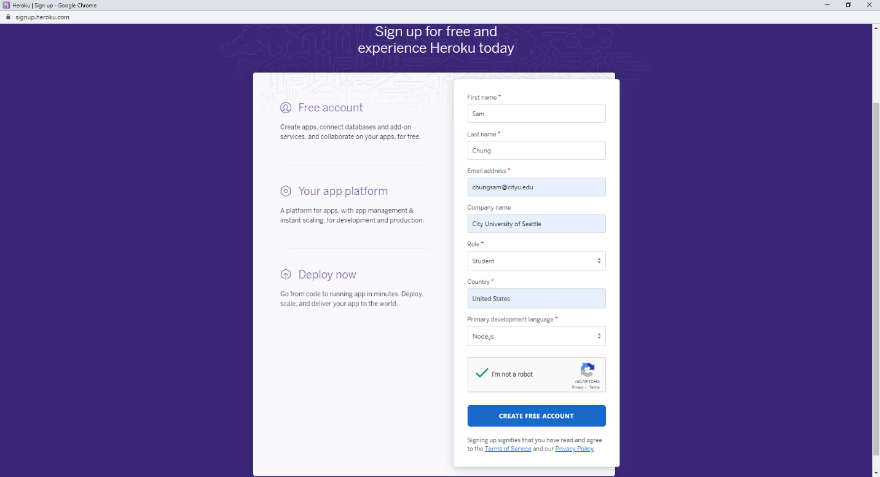
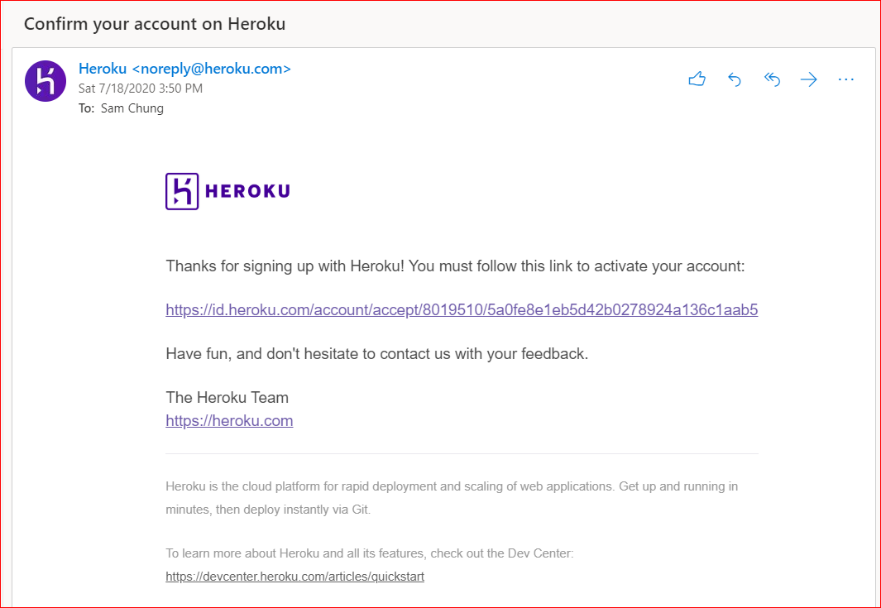
**Creating an Express Project with Scaffolds**

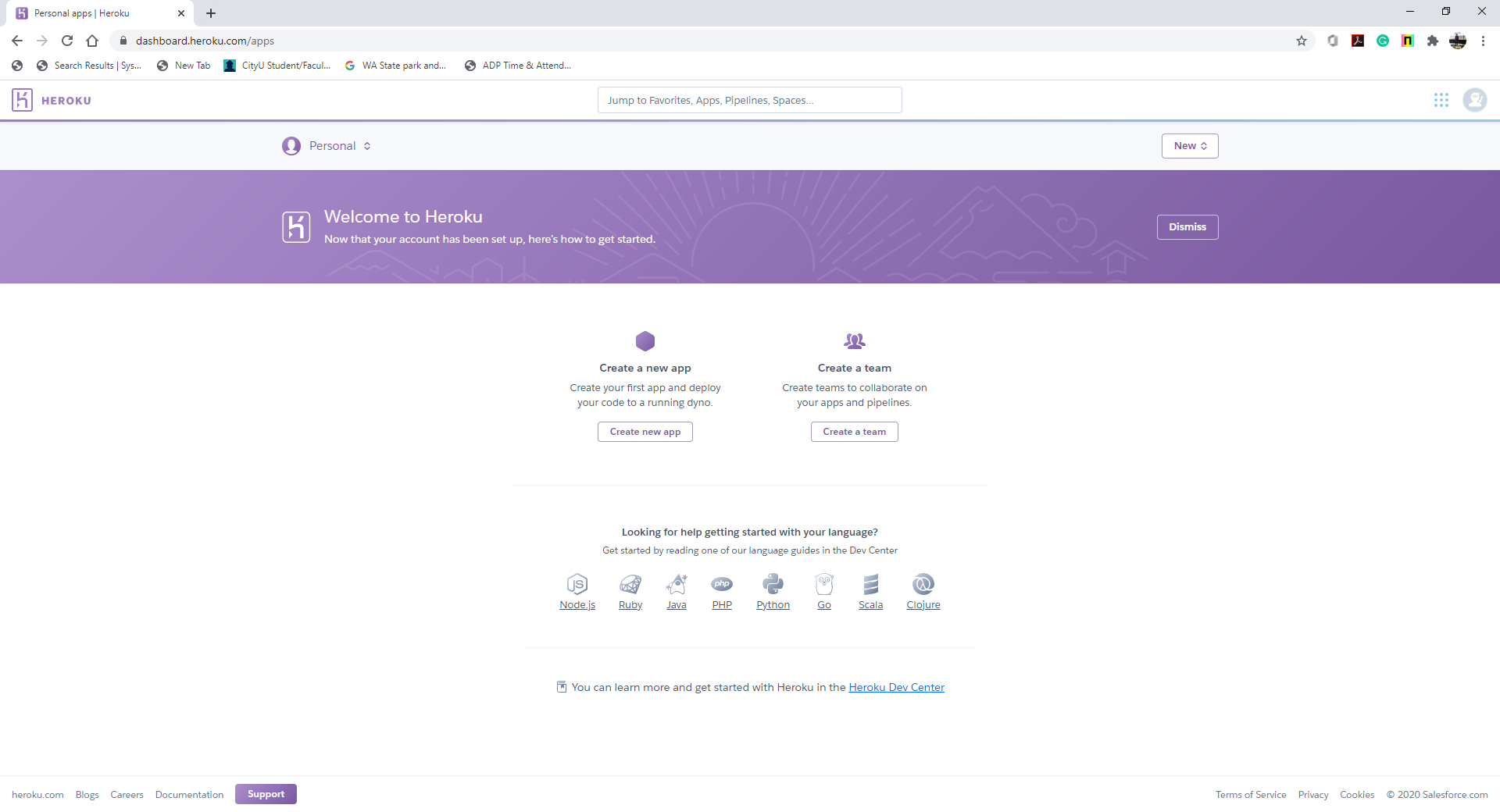
1. You are going to use the Pug template engine for HTML in the express framework.   
   You will also let it generate a “.gitignore” file for you.  
     
   In the VS code terminal type the following command:  
   

This command creates a bunch of folders and files (i.e. scaffold, scaffolding or staging) inside the module03a folder that form the basis of the application. Your folders inside the module03a should look like this.  
  


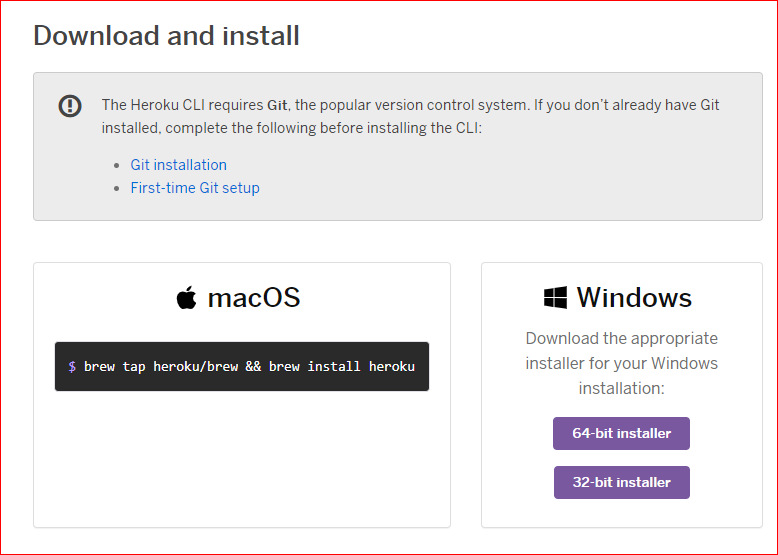
1. Next, you need to install the dependencies. As you may remember, you do this by running the following command from a terminal prompt in the same folder as the “package.json” file:  
   >>>npm install  
     
     
   If you found any severity vulnerability,  
   >>> npm audit fix
2. Start your Express application.  
   >>> npm start  
   
3. To be able to see the GET requests, you have to go to <http://localhost:3000/> in your browser.  
   
4. To stop your Express server, hit Ctrl+C.  
   
5. Try to access <http://localhost:3000/> in your browser again. You should see the following:  
   

**Deploying Your Project onto a Cloud PaaS**

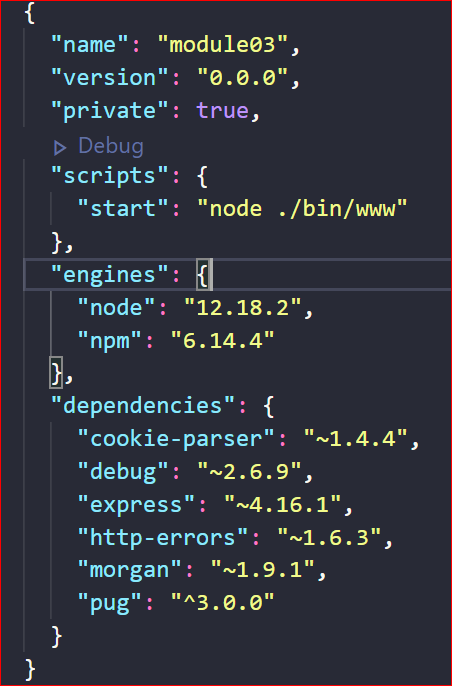
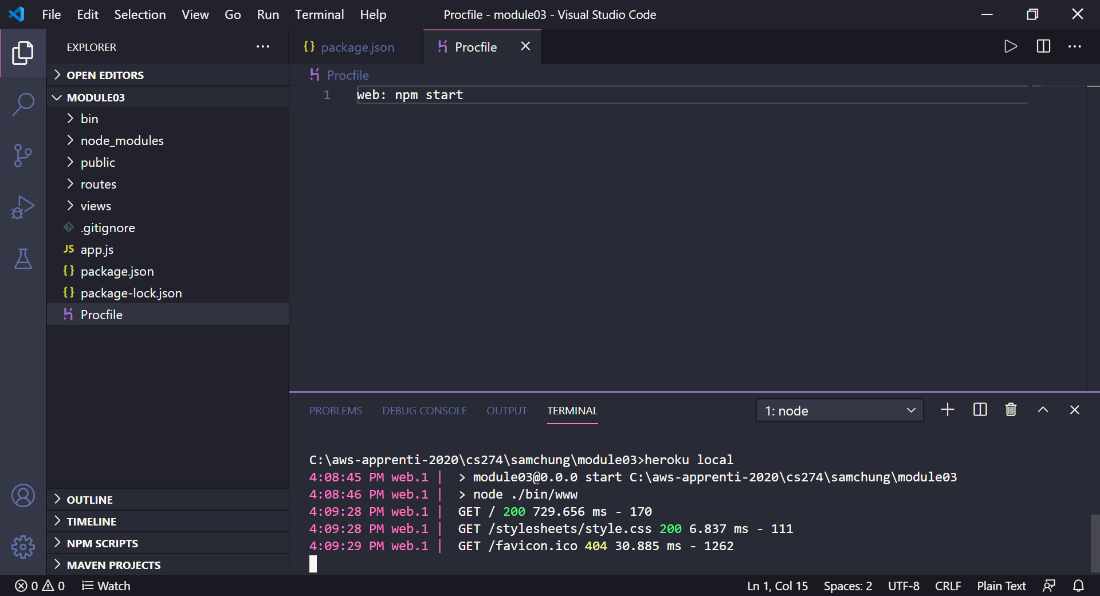
1. Let’s sign up to Heroku: <https://id.heroku.com/login>.  
   Heroku is a cloud platform as a service (PaaS) supporting several programming languages.   
   We choose Heroku for ease of setup and to follow along with the textbook.  
     
     
   

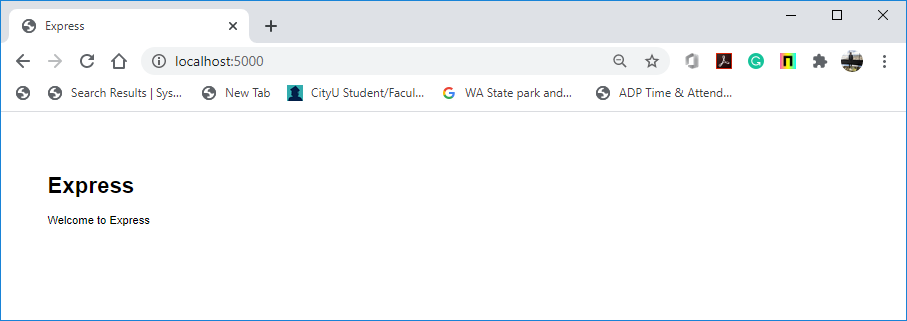


1. Download and install the Heroku CLI.
   1. You will use it from terminal to manage your Heroku deployment.
   2. Local is useful for making sure that what you have built on your machine is set up to run properly on Heroku.



Note: If you cannot find the page, you can access via https://devcenter.heroku.com/articles/getting-started-with-nodejs#set-up

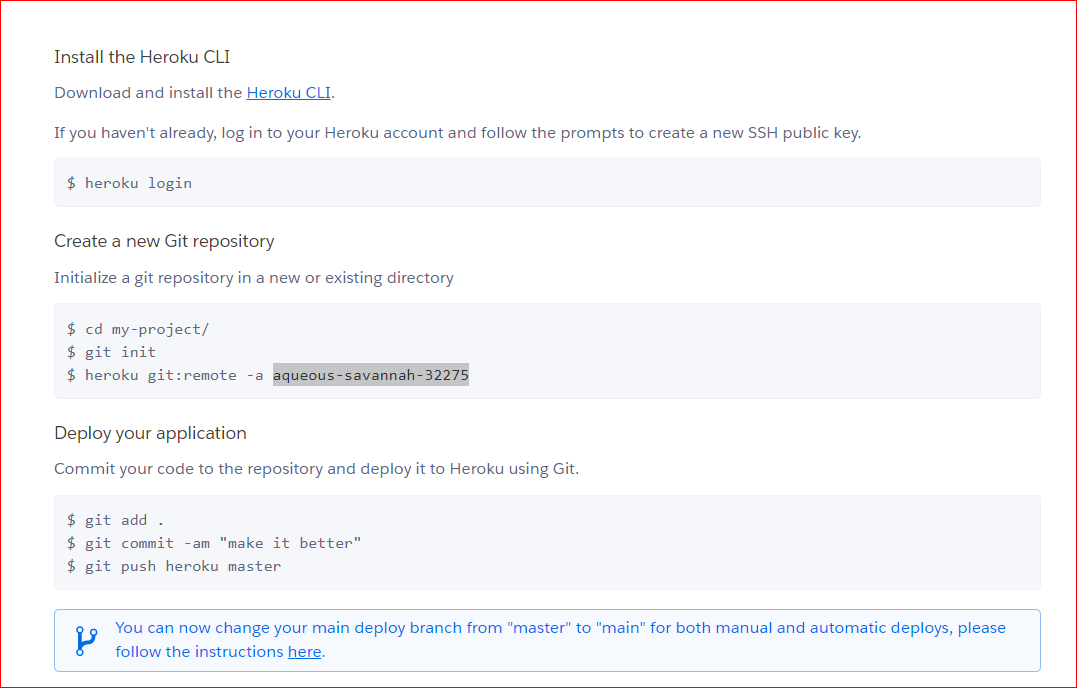
1. Update the package.json to let the Heorku know which versions of node and npm your application is using for the development and the production set up.  
   
2. Declare the process types used by your application to tell Heroku how to start the application.  
   Create a **Procfile** in the root directory of your project folder.  
   Add the following:  
   *web: npm start*
3. Test your Heroku setup and run your application locally before pushing the application up to Heroku.  
   >>>heroku local  
   
4. Access Heroku in your local machine with: <http://localhost:5000/>  
   The local test app’s port number is 5000.



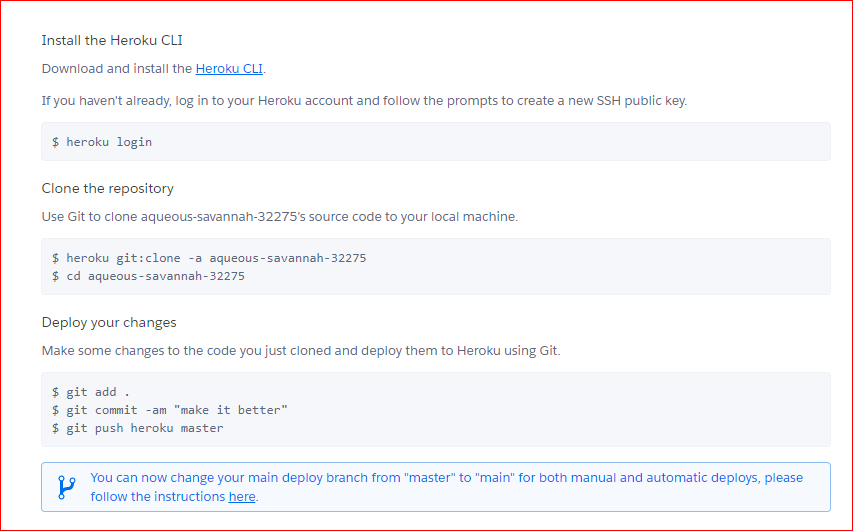
1. Deploy your application up to the cloud PaaS Heroku.

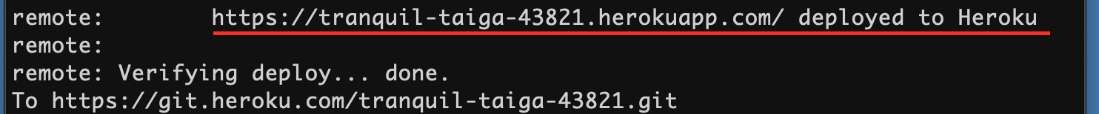
Commit your changes  
>>>git init  
>>>git add --all  
>>>git commit –m “First Commit”  
  
Create the app  
>>>heroku create  
*Note: it might ask you to sign up*>>>git push heroku *your\_current\_branch*

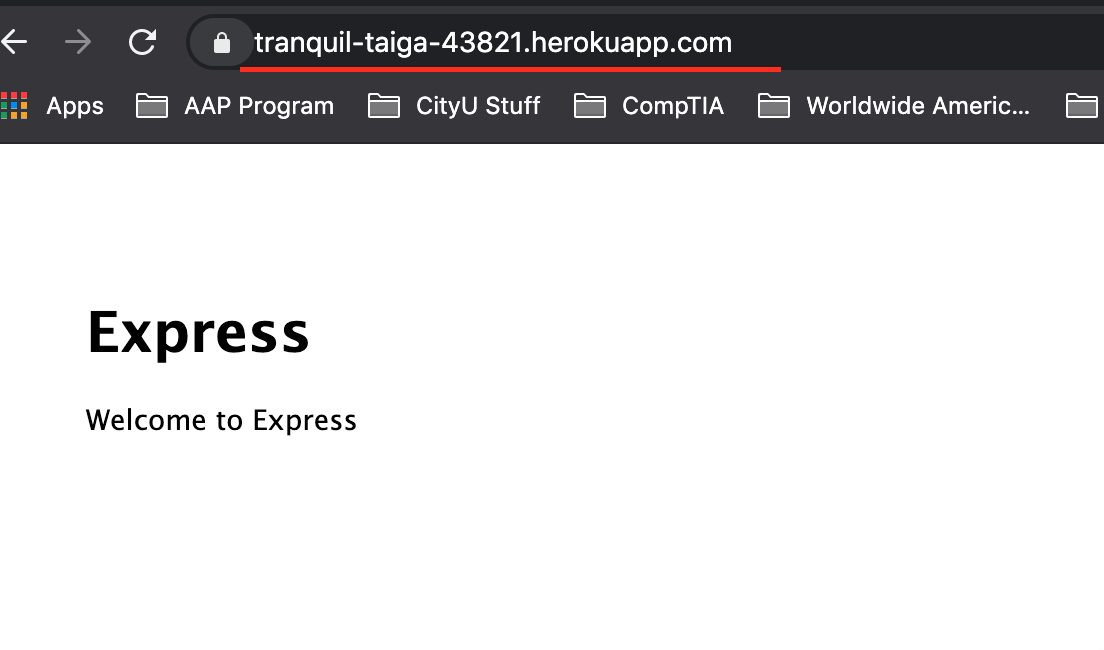
After you use “>>>heroku create”

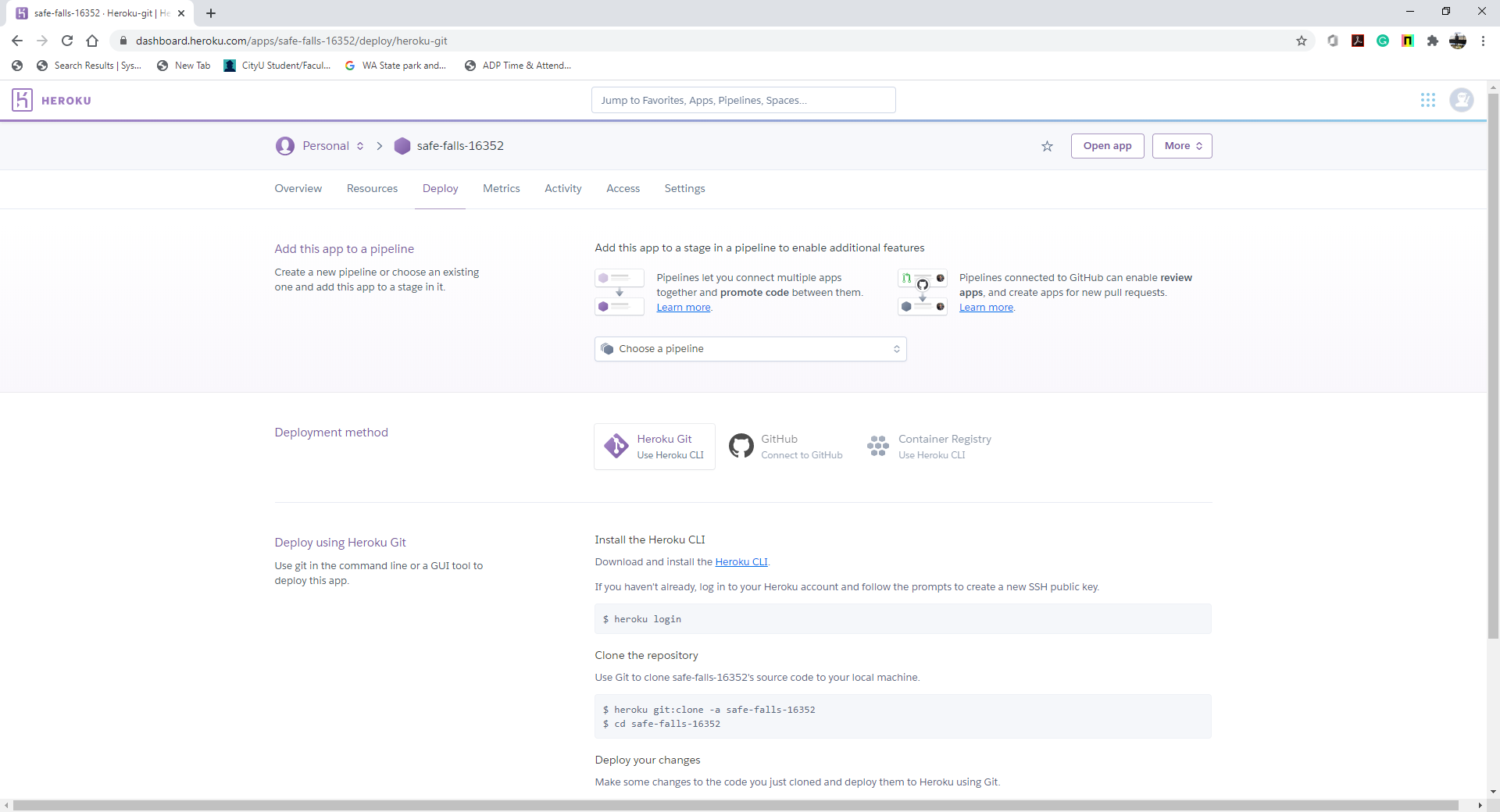


For the next deployment after the first creation,



To find the Heroku App url, check the output:  




If you have any deployment issue, visit your application in Heroku.  


**Source Code Management and Deployment**

**Push your work to GitHub**

Run the following commands to push your work to the GitHub repository:

Open the terminal from the VSCode by hitting the control + ~ key and type the following command:

>>> git add .

>>> git commit -m “Submission for Module 3 –yourname”

>>> git push origin master

**Push your work to Heroku**

>>> git push heroku master

If you cannot remember your branch name, run the command “git status” to check.