Team Waffle io board link :- <https://waffle.io/nehaparmarcs/ShoppingCart>

Team GitHub repository :- https://github.com/nehaparmarcs/ShoppingCart

Navdeep Patel

Github username- navdeep2237

(Team\_10)

This is a very interesting project in which we will have a chance to learn a lot about NoSQL databases and face the challenges in implementing them. Our team has decided to use JAVA as the programming language for the backend and hopefully there would be usage of Node.js as well so, I have started learning Node.js. We have decided upon CouchDB for the shopping catalogue and Riak for the shopping cart and there could be more additions if time permits.

We had a team meeting on Wednesday in which we decided about the products on which we will be basing our e-commerce website on. We also decided that 2 people will be working on Frontend(including me) and 3 people would be working on the backend. But since this is a group project and everybody should have a working knowledge of the project and how things work in the project, we will be constantly updating our team members about the progress we are making.

I have installed Heroku toolbelt for windows with which I would be able to deploy an application using powershell or command prompt. I have also have gone through the tutorial provided at <https://devcenter.heroku.com/start> to deploy a sample java and a Node.js application on Heroku platform. In the tutorial I learned about *Procfile* (A text file in the root directory of the application which explicitly tells which commands need to be executed to start the app) and how to change the *Procfile* so that we can deploy an app from windows. I also learned how to add dependencies for *Maven* and how to install *Maven*(used command *.\mvnw clean install* as I was using PowerShell). I learned a way to scale up and scale down an application on Heroku(i.e how many dynos we are using). The Java application was prebuilt and is available on github.com ( <https://github.com/heroku/java-getting-started.git> ) . I learned a few commands for using github .

I also Deployed a sample Node.js application on Heroku using similar steps as the deployment of java application( the screen shots are available on github ) but needs *Node.js and npm* to be installed locally on our machines.

***Requirements:***

*Java8* and *Maven8* installed on our local machine

***Step 1***: *Setup* –

Install Heroku tool belt which provides a command line interface for windows to help manage the applications and add-ons.

After the installation is complete, we login to our Heroku account from PowerShell using the command

*$heroku login*

***Step 2****- preparing the app*-

In this step we clone a repository with the “*git clone [url]* “command which contains the sample java application.

***Step 3*** *– Deploying the app –*

We use the command *$heroku create* . This command creates a git remote (called heroku) and it is associated with our local git repository.

After the git remote is added, we use the command

*$git push heroku master* to deploy our code.

To ensure that atleast one instance of our app is running we use the command

*$heroku ps:scale web=1*

Now, we can open the application either with the url which was provided at the end of git push to the master or we can use the command

*$heroku open* – this command opens a new tab with our deployed java application.

***Step 4****- View Logs-*

We can view the logs about our application using the command

*$heroku logs –tail*

These logs are time ordered events and with every visit to the page, the logs are updated.

***Step 6****- Defining a procfile.*

A Procfile is a text file which is the root directory of our application and this file explicitly declares the commands which have to be used to start our app.

For windows we can add a line into our Procfile :

*web: java –cp target\classess;target\dependency\\* Main*

***Step 7****- Scaling –*

When we scale an app on heroku , we change the number of dynos which are running.

So to scale up we use the command *$heroku ps:scale web=1*

and to scale down we use the command *$heroku ps:scale web=1*

***Step 8****- Declaring the app dependencies.*

When we have java application , Heroku recognizes it by the pom.xml in the root directory.

For installing the dependencies we use the command

*$mvn clean install* ( we need Maven already installed on the local machine for this to run) or for windows we need to run *mvnw clean install.*

***Step 9****- Running the app locally*-

We use the command *$heroku local web* ( for running it on windows we need to append *–f* *Procfile.windows*) . This command will start our app on <http://localhost:5000> .

***Step 10*** *– Pushing local changes*

In this part, I modified pom.xml to include the dependency for *jscience* and then modified src/main/java/main.java to include *org.jscience.physics.model.RelativisticModel* and *org.jscience.physics.amount.Amount*. I tested it locally first.

Then, I used the commands *git add .* *, git commit –m “ “* and *git push heroku master* to put it on the github repository.

For provisioning an add-on we need to have our accounts verified which needs providing credit card details.

Heroku has Postgres provisioned automatically on all deployments of java(free). Databases are add ons in Heroku ( Redis, MongoDB, Postgres, MySQL etc.).

During the tutorial on Heroku, I was able to deploy the sample application locally and online. But after I made a few changes in the program, I was able to deploy the application locally but was unable to deploy the application online for some time due to an error I made in the program, this taught me to be aware about this kind of simple mistakes and double check the code before deploying .

With the understanding of how to deploy a java and a node.js application on Heroku, my next agenda is to get to know more about Heroku and java together and also to start learning about *Spring MVC Hibernate*.

Heroku supports 5 maximum number of apps before it asks for verification (entering the credit card details). And for adding the add-ons, there is a verification step as well.

I have also started learning the basics about CouchDB and Riak as they will be used as databases for the catalogue and the shopping cart and also for the upcoming Presentation on the 16th April.