CSC301 - Assignment 1

Kaushik Murali and Hyeon Jeong Byeon

Summary

Group: assignment-1-81-kaushikmurali01-hyeonjeongbyeon

Assignment Repository

https://github.com/csc301-fall-2021/assignment-1-81-kaushikmurali01-hyeonjeongbyeon

Environments

(1) Development Environment:
http://checkoutcalculatortesting-env.eba-3imhvu24.us-east-2.elasticbeanstalk.com/

(2) Production Environment:
Checkoutcalculator-env.eba-s2eq4spd.us-east-2.elasticbeanstalk.com

Video rundown explaining functionality of web app

https://www.youtube.com/watch?v=UjxIrOY6X0A&ab_channel=XOsKeLeToN

Video explaining CICD and deployment

https://www.youtube.com/watch?v=tJop3hY5gZI&ab_channel=XOsKeLeToN

In assignment 1, we chose technologies that we were somewhat familiar with from our prior experience in web app development as well as technologies that we believe might be helpful in preparing us for the team project.

We used the following technologies:

Technologies:

- Frontend: HTML, CSS, Javascript and Jquery
- Backend: Java and JSP
- CI/CD: Jenkins
- Hosting: AWS Elastic Beanstalk
- Tomcat servlet engine was also used for testing on localhost

In our decision making process to pick the technologies we wanted to use to make our web application, we examined many options for 5 main components. The components were frontend, backend, CI/CD, Hosting and Database.

Frontend

For the frontend, we considered many options but we narrowed it down to three options. The options were React, Vue and finally [html, css, javascript and jquery]. We considered the pros and cons of each before making our choice.

React is easy to learn, provides for adapting the web app for different platforms and is SEO friendly. However, it has poor documentation and a fast development pace making it hard for developers to get used to the environment.

Vue is an easy to learn language, has two way communication between UI and model data, and has Good virtual dom rendering and performance. However, it is too flexible because sometimes giving developers too many options may cause errors.

[html, css, javascript and jquery] makes code simple and readable with jquery. There are easy to learn languages. We have a lot of prior knowledge and experience in using them. However, the con is that jquery functionality is limited despite the big library.

We chose the third option **[html, css, javascript and jquery]** because we were already quite familiar with using these and have taken online web development courses that used these languages. Also, it is good for small web applications like the checkout calculator web app.

Backend

Similarly, for the backend we considered python django, NodeJs Express and finally java and jsp.

Python Django is easy to learn, has a fast development process and provides a wide range of libraries. However, complex programs may take longer than normal to execute and code written in one version may not work in another version.

NodeJs Express provides high performance and Object-Oriented Programming. It also requires no need to learn another server-side language. However, it lacks an extensive library support,

Java and jsp have a free and very good IDE that is available. It is an object-oriented programming language. It supports multithreading. We also have a lot of experience using this. However, the con is that the programming language is time-consuming.

We chose the third option **Java and jsp** because we were already quite familiar with using these and have taken online web development courses that used these languages, the Syntax is simple to understand, it is an object-oriented programming language and it also enables the server side to run many instances at once.

CI/CD

We considered Github Action, CircleCI and Jenkins.

Github Actions is free, it provides us with access to github api without providing authentication and it is easy to configure. However, it does not have that many plugins.

CircleCI is Very configurable, requires no need for maintaining your own infrastructure and has a very huge user base for support. However, it also doesn't have that many plugins.

Jenkins is very customizable as there are a lot of plugins. It is free and open source and has simple architecture to deploy. It also has a large community to provide support since it is widely used. However, it also has a lot of redundant plugins that we probably will not use.

We chose the third option, **Jenkins**, because it is very customizable and configurable due to the extensive plugins which simplifies the CI/CD flow. It has a large community to provide support if I do not know how to do something and also a nice UI making configuring projects very easy to do.

Database

We considered MySQL, MongoDB, PostgreSQL and also no database at all (since it was optional).

MySQL is very fast and reliable and stores relational data. It was made paying attention to web and Big Data. It has high security and supports large databases. However, it is complex and time consuming to design on top of not having a very good debugging tool.

PostgreSQL has a lot of predefined functions. It has many available interfaces and it also supports json. However, configuration is hard to understand and perform.

MongoDB has good performance. It is easy to set up. It is a horizontally scalable database. It is also suitable for hierarchical data storage. However, it does not support transactions.

No database - maintained in memory is Easy and less work. A database is not required for this assignment. No database is required for small scale projects such as a checkout calculator. However, the con is when you add items to the product list, once the server is shut down, the added items are lost.

We chose the fourth option, **no database - maintained in memory**, because it is easy to implement and unnecessary for small scale web applications. Database is not required by the assignment and creating a database is complex and time consuming.

Web Hosting Service

We considered Heroku, Azure Web Apps and AWS Elastic Beanstalk.

Heroku is free, suitable for small apps and easy to scale. However, its performance is not as good compared to some others.

Azure Web Apps has very good scalability, has a good management dashboard and is fast and easy to set up. However, it might be a little hard to initially learn and get used to

AWS Elastic Beanstalk has easy to build infrastructure, is easy to deploy and has good auto-scaling settings. However, it can sometimes be slow.

We chose the third option, **AWS Elastic Beanstalk**, because it is has a free tier available, the deployment process is very simple, it has very good customization and also because Elastic beanstalk supports Java and provides Tomcat environment which is necessary for our web application.