BEng(H) in Software & Electronic Engineering

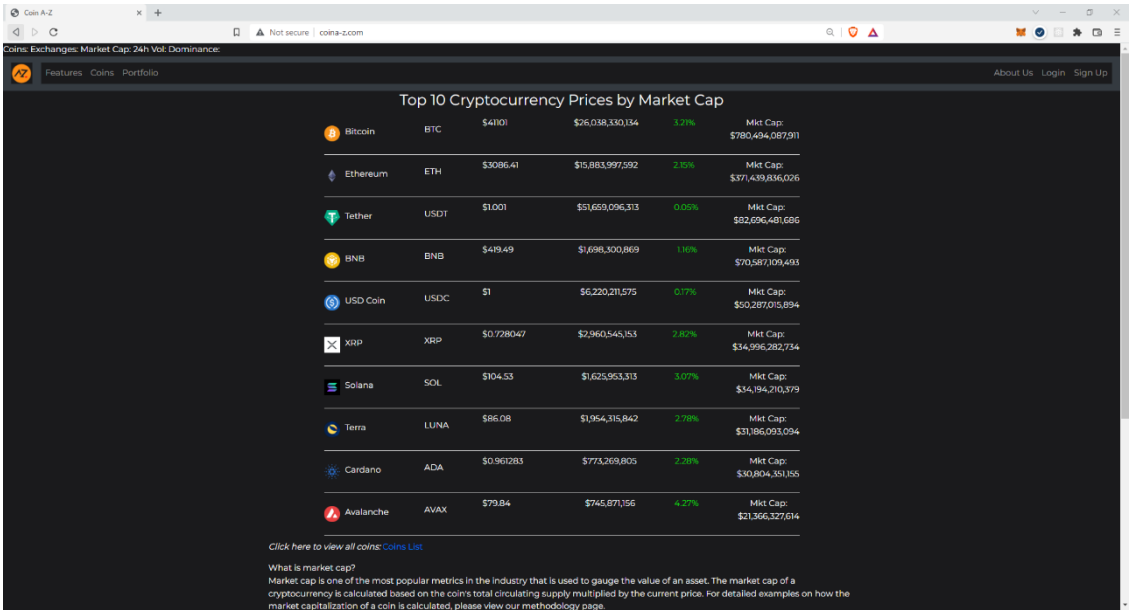
Project Engineering

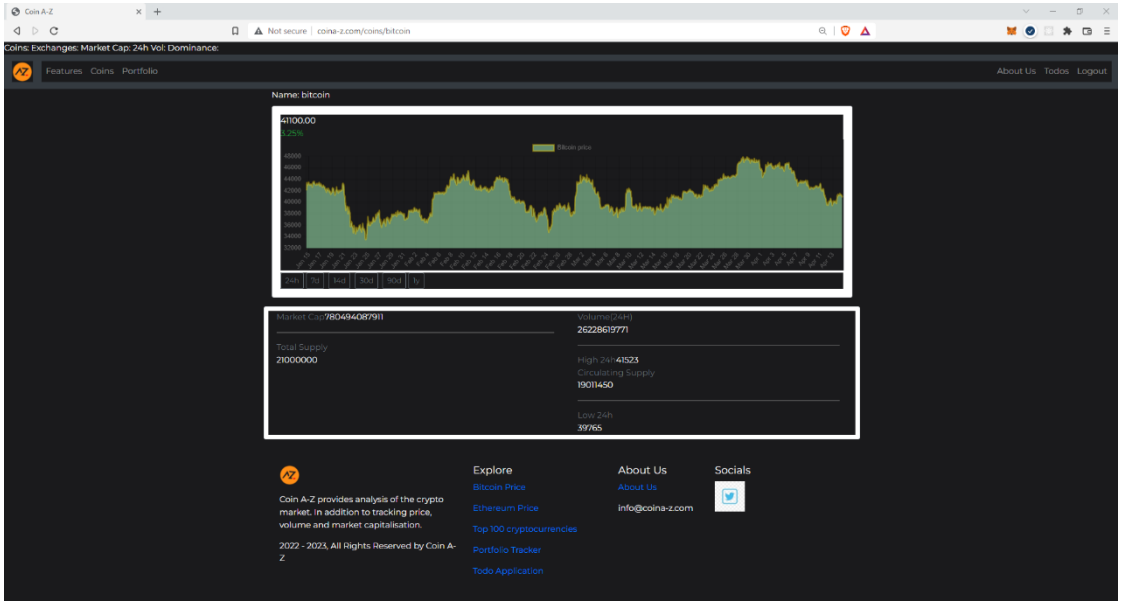
Coin A-Z

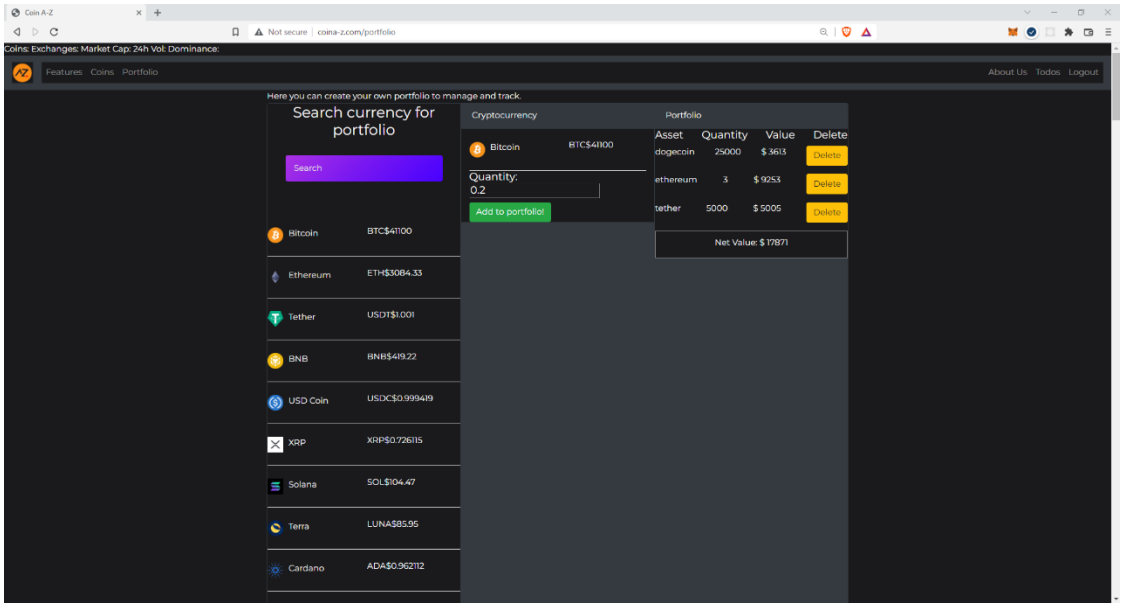
Niall O'Keeffe

Year 4

Ciarán Slattery - G00373551


Coin A-Z
Not secure | coina-z.com
Coins: Exchanges: Market Cap: 24h Vol: Dominance:
Features Coins Portfolio
About Us Login Sign Up
Top 10 Cryptocurrency Prices by Market Cap
Bitcoin BTC $41101 $26,038,330,134 3.21% Mkt Cap: $780,494,087,911
Ethereum ETH $3086.41 $15,883,997,592 2.15% Mkt Cap: $371,439,836,026
Tether USDT $1.001 $51,659,096,313 0.05% Mkt Cap: $82,696,481,686
BNB BNB $419.49 $1,698,300,869 1.16% Mkt Cap: $70,587,109,493
USD Coin USDC $1 $6,220,211,575 0.17% Mkt Cap: $50,287,015,894
XRP XRP $0.728047 $2,960,545,153 2.82% Mkt Cap: $34,996,282,734
Solana SOL $104.53 $1,625,953,313 3.07% Mkt Cap: $34,194,210,379
Terra LUNA $86.08 $1,954,315,842 2.78% Mkt Cap: $31,186,093,094
Cardano ADA $0.961283 $773,269,805 2.28% Mkt Cap: $30,804,351,155
Avalanche AVAX $79.84 $745,871,156 4.27% Mkt Cap: $21,366,327,614
Click here to view all coins! Coins List
What is market cap?
Market cap is one of the most popular metrics in the industry that is used to gauge the value of an asset. The market cap of a cryptocurrency is calculated based on the coin's total circulating supply multiplied by the current price. For detailed examples on how the market capitalization of a coin is calculated, please view our methodology page.


Coin A-Z
Not secure | coina-z.com/coins/bitcoin
Coins: Exchanges: Market Cap: 24h Vol: Dominance:
Features Coins Portfolio
About Us Todos Logout
Name: bitcoin
41100.00
1.25%
Bitcoin price
24h 7d 1m 30d 90d 1y
Market Cap: 780494087911
Volume(24H): 26228619771
Total Supply: 21000000
High 24h: 41523
Circulating Supply: 19011450
Low 24h: 39765
Coin A-Z provides analysis of the crypto market, in addition to tracking price, volume and market capitalisation.
2022 - 2023, All Rights Reserved by Coin A-Z
Explore
Bitcoin Price
Ethereum Price
Top 100 cryptocurrencies
Portfolio Tracker
Todo Application
About Us
About Us
info@coina-z.com
Socials


Coin A-Z
Not secure | coina-z.com/portfolio
Coins: Exchanges: Market Cap: 24h Vol: Dominance:
Features Coins Portfolio
About Us Todos Logout
Here you can create your own portfolio to manage and track.
Search currency for portfolio
Search
Bitcoin BTC$41100
Ethereum ETH$3084.33
Tether USDT$1.001
BNB BNB$419.22
USD Coin USDC$0.999419
XRP XRP$0.726115
Solana SOL$104.47
Terra LUNA$85.95
Cardano ADA$0.962112
Cryptocurrency
Bitcoin BTC$41100
Quantity: 0.2
Add to portfolio!
Portfolio
Asset Quantity Value Delete
dogecoin 25000 $ 3613 Delete
ethereum 3 $ 9253 Delete
tether 5000 $ 5005 Delete
Net Value: $ 17871

Declaration

This project is presented in partial fulfilment of the requirements for the degree of Bachelor of Engineering (Honours) in Software and Electronic Engineering at Galway-Mayo Institute of Technology.

This project is my own work, except where otherwise accredited. Where the work of others has been used or incorporated during this project, this is acknowledged and referenced.

Table of Contents

1 Summary

Coin A-Z is a cryptocurrency tracking application that allows users to monitor coins of interest on the market. The user can search specific coins and get all the historical data along with the current data about the coin including charts. The user can create a custom portfolio to track their entire portfolio value in one place along with tracking of individual assets.

1.1 Goal

My aim for my FYP is to design a full-stack cryptocurrency tracking application. The functionality I wish to provide will include:

- Allow user to search for coins
- Allow user to create a custom portfolio
- Allow user to look at coin data and historical charts

I am hoping to implement such elements through learning new frontend and backend frameworks, while also incorporating any knowledge that we have attained throughout our undergraduate studies, such as ReactJS, Java, HTML and MySQL.

1.2 Scope

In terms of implementation, my goal is to learn new and industry relevant frameworks, which will allow me to gain knowledge in areas that I have not been shown throughout my academic studies. For the frontend of the application, I will use ReactJS, a JavaScript framework. Using React will also allow me to incorporate my knowledge of HTML, CSS and Bootstrap to design and style all web pages. For the backend I will use the Spring Framework which is written in Java and will allow a means of communicating with the database.

1.3 Approach

I have approached Coin A-Z in an agile methodology, I set out guidelines which I would work on in similar spacing timelines. I broke up my project into different tasks which I planned to tackle one by one.

1.4 Accomplished

I have accomplished all that I had set out to develop at the start of my project planning. I feel I have efficiently developed Coin A-Z in my time frame.

Coin A-Z

Full-Stack cryptocurrency tracking and portfolio manager webapp - coina-z.com

Ollscoil Teicneolaíochta an Atlantaigh
Atlantic Technological University

Ciarán Slattery
BEng (H) Software and Electronic Engineering 2021/22
Supervisor: Niall O'Keeffe
Project Engineering

Introduction - Coin A-Z

Description:

Coin A-Z is a cryptocurrency tracking application that allows users to monitor coins of interest on the market.
The user can search specific coins and get all the historical data along with the current data about the coin including charts.
The user can create a custom portfolio to track their entire portfolio value in one place along with tracking of individual assets.

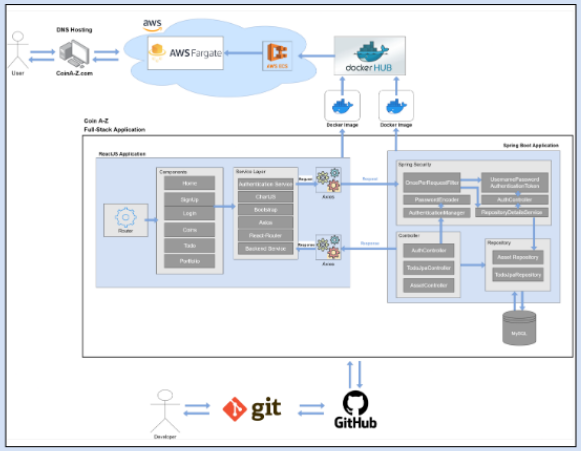
Goals:

- A scalable micro-service based web application hosted on AWS.
- Fast and responsive web-app.
- User has an easy-to-use user interface where they can have all their analytic needs in one site.
- User is able to register an account
- User is able to create a portfolio of coins which will allow them to track the price of their entire portfolio and individual assets.
- User will have access to historical data on coins along with charts.
- Secure login for users using JWT (Jason Web Token).

Technologies

- Coin A-Z uses microservices to deliver any coin request that the user might input. It includes Spring Boot REST applications, ReactJS UI and MySQL, it uses Docker to wrap the applications in containers. The containers will simplify the delivery and management of these microservices while also making the applications scalable.
- The project/app retrieves the cryptocurrency data and analytics using the CoinGeckoAPI.
- The front-end is developed with React and JavaScript to display the User Interface and handle the user's inputs. It uses many different React libraries such as Chart.js to chart the coin graphs, Axios for URL requests and Bootstrap. Coin A-Z has an easy-to-use user interface with useful features.
- The project uses MySQL for storing user's data such as their information and their personal portfolio.
- The entire application is hosted by Amazon Web Services using ECS. Using ECS makes it easy to deploy my docker images directly without any configuration on a virtual EC2.
- The site itself uses DNS on a domain from godaddy.com.

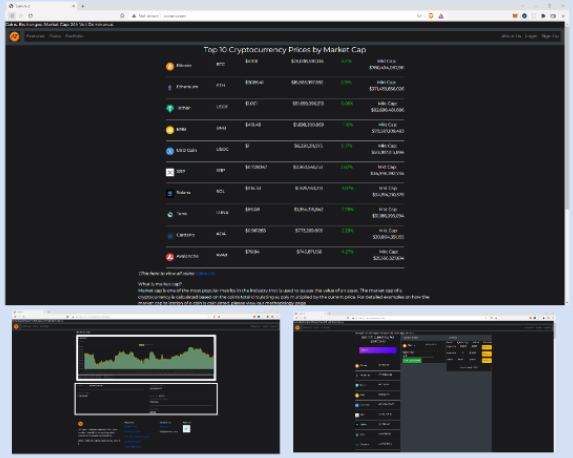
Architecture Diagram



Functionalities & Features

- Full-Stack functionality
- Micro-serviced based
- ChartJS (Coin Charts)
- Axios (URL Requests)
- API Requests (CoinGeckoApi)
- User registration
- User portfolio tracker
- Individual asset tracking
- User task creator
- JWT
- Dockerized
- AWS Hosted (ECS/EC2)
- DNS

Results



Conclusion

For my final year project, I set out to design a cryptocurrency tracking application and website using React, Springboot, MySQL, Docker and AWS. The website would allow a user to create an account, build a custom portfolio of coins and track the their asset value. Users will also be able to look at coin data and graphs. I designed and built from scratch a fully functional full-stack website which met the requirements I originally set out to complete. Using languages such as Java, JavaScript, HTML, CSS, Bootstrap, SQL allowed me to develop Coin A-Z, which allows users to easily understand and use the system without having ever used it before, with ease and efficiency. Using AWS, I hosted my frontend and backend applications, I also used DNS to host my website on a domain I acquired.

Having gone through the process of developing a software project from start to finish, I have enhanced my overall knowledge of each area SDLC (Software Design Life Cycle), which also incorporating and building on elements of modules that I have studied in my undergraduate course. The process has gave me an understanding of the all hard work and planning that goes into software development projects and has allowed me to learn how to develop a full-stack application, which is valuable knowledge I will be taking with me further in my software career moving forward.

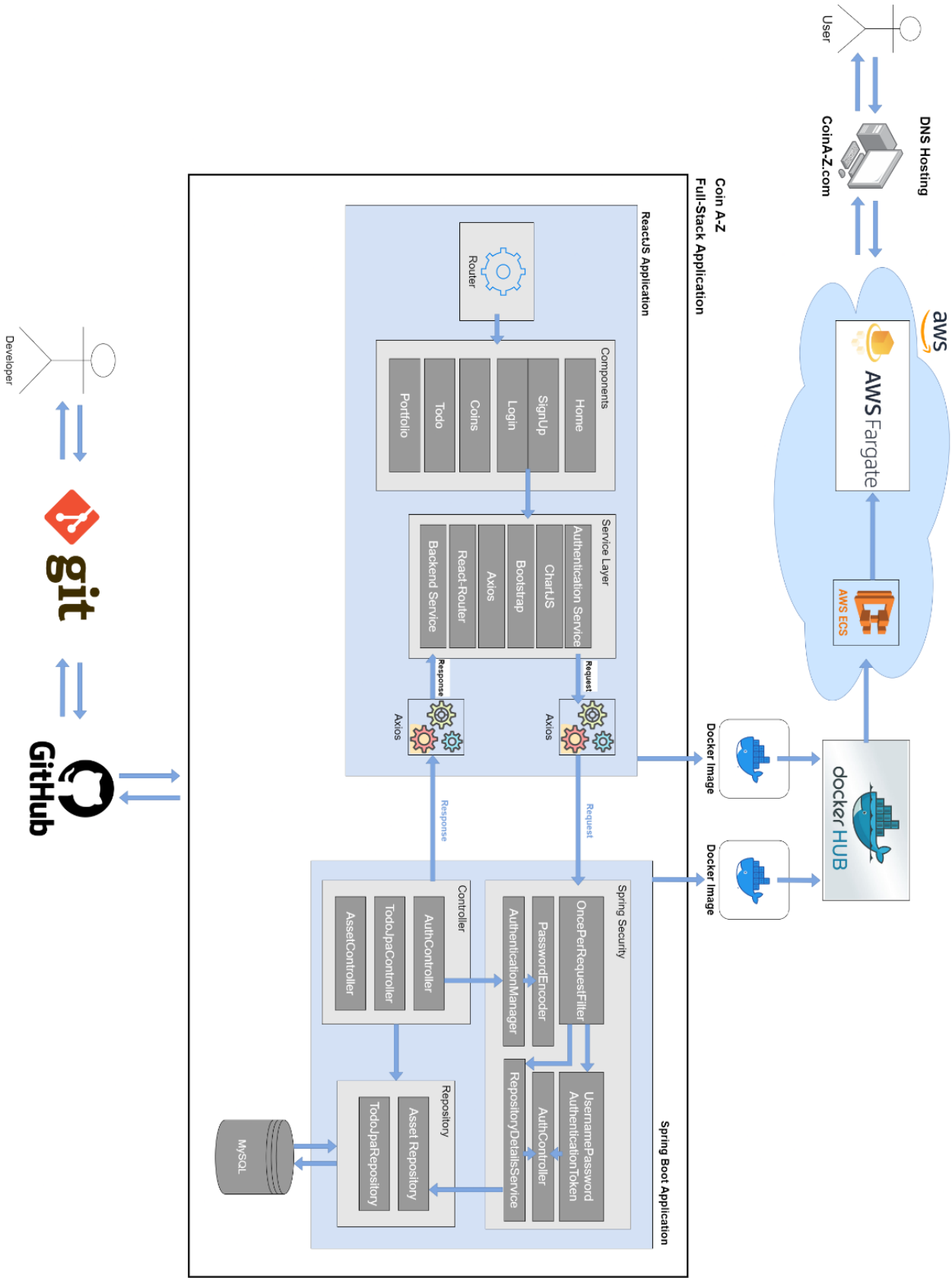
https://github.com/CiaranSlattery/CoinA-Z

3 Introduction

Coin A-Z is a cryptocurrency tracking application that allows users to monitor updates and trends relating to coins of interest on the market. The user can search specific coins and get all historical along with the current data about the coin including charts. The user will have an easy-to-use user interface where they can have all their analytic needs in one site. A user will be able to register an account and create a portfolio of coins which will allow them to track the price of their entire portfolio and individual assets. Coin A-Z is a scalable micro-service-based web application.

Coin A-Z uses microservices to deliver any price request that the user might input. It includes Spring Boot REST applications and uses Docker to wrap these applications in containers. The containers will simplify the delivery and management of these microservices while also making the applications scalable. The project/app will retrieve the crypto coin data and analytics using the CoinGeckoAPI. The front-end is developed with React and JavaScript to display the User Interface and manage the user's inputs. Coin A-Z has an easy-to-use user interface with useful features. Coin A-Z uses MySQL for storing user's data such as their information and their personal portfolio. Amazon Web Services are hosting the entire application.

4 Project Architecture



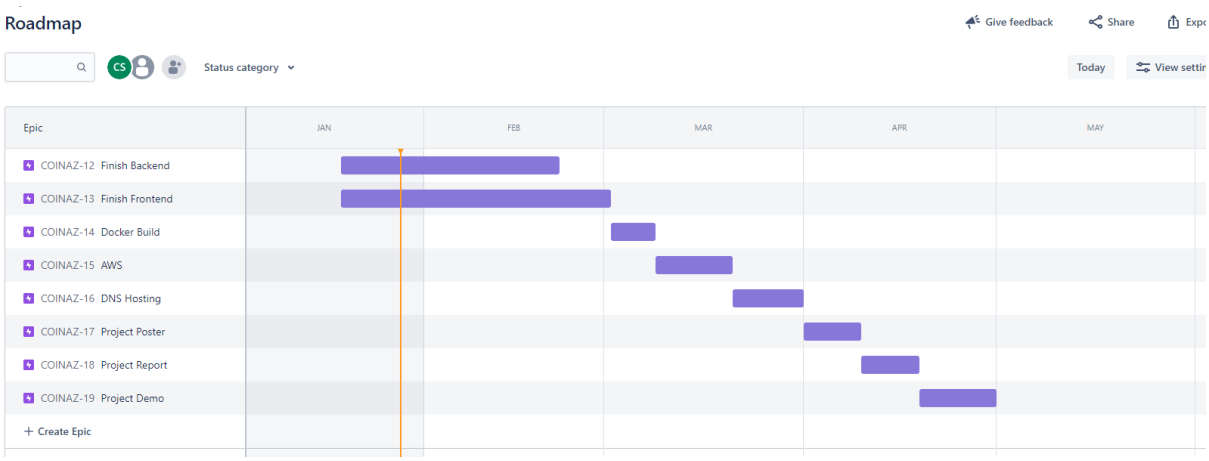
5 Project Plan

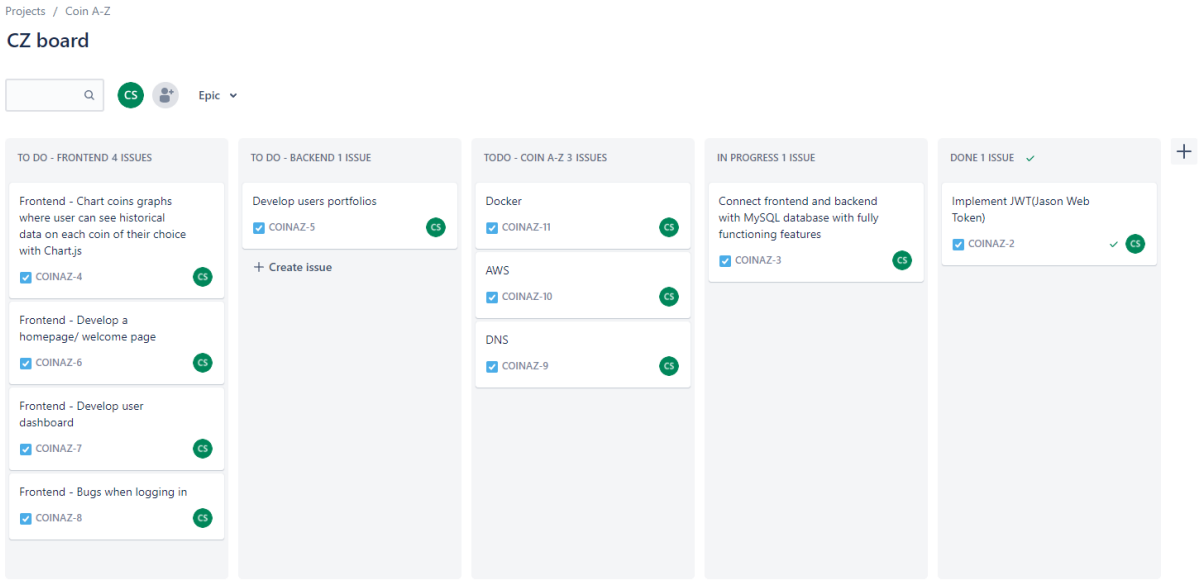
Jira

Jira is a proprietary issue tracking product developed by Atlassian that allows bug tracking and agile project management. (Jira, 2022)

Using Jira software, I mapped out my project plan and organized my project with a Gantt chart.

Using Jira boards also allows you to track tasks and assign tasks to your team, I used the board for organizing all my tasks efficiently.

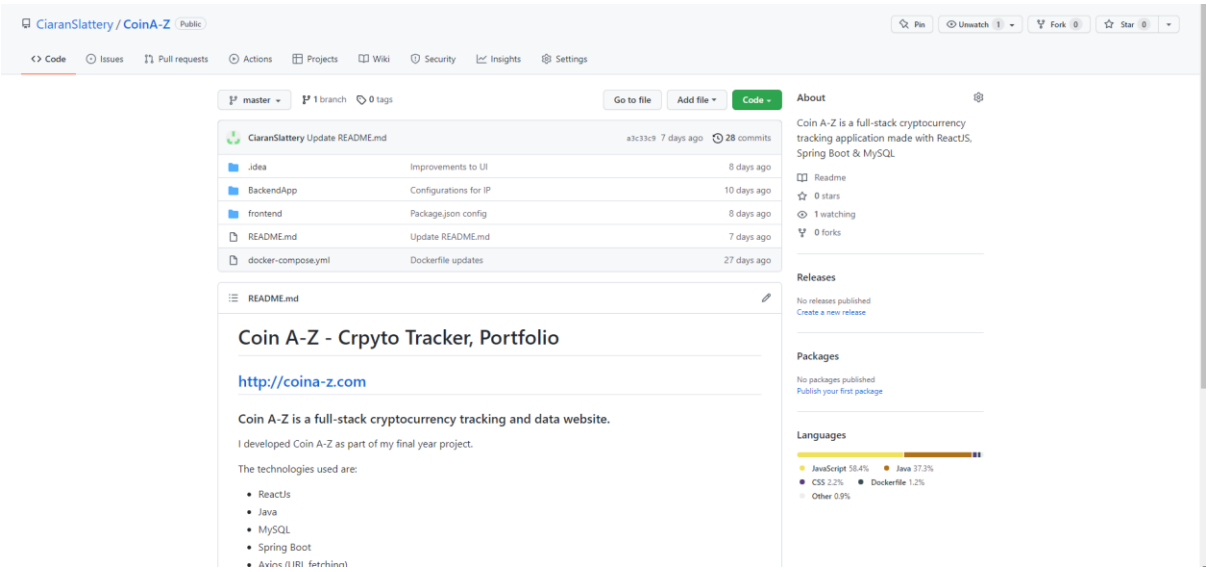




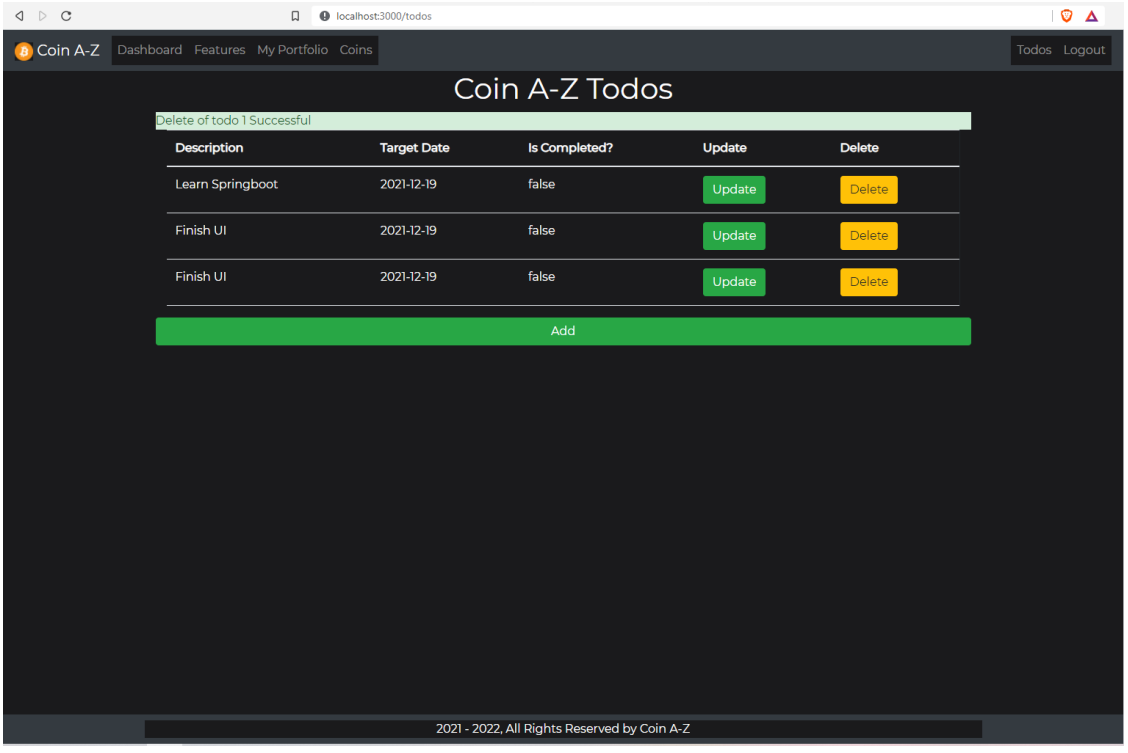
GitHub

GitHub, Inc. is a provider of Internet hosting for software development and version control using Git. It offers the distributed version control and source code management functionality of Git. (GitHub, 2022)

Using GitHub, I was able to keep my code organized and easy to access along with different versions through git commits.



I also have a to do application implemented into my project itself. I found it a handy feature and able to update the tasks with my needs.



6 Technologies & Code

The design pattern of the application is Model-View-Controller design pattern. This will provide the low coupling between the model, view and controller and will allow the application to be more scalable and maintainable.

The model component is the MySQL database holding and managing the data. The view component is the user interface which will render presentations of the model to the user. It will also allow for user interaction. The controller component is the backend which will respond to user requests triggered by the view component and will interact with the model component to provide a response the view will render. This will allow the controller to be reusable as well.

I based my project on a tutorial course and used it for a base to develop Coin A-Z: https://github.com/in28minutes/full-stack-with-react-and-spring-boot

Frontend

For the frontend development I used the Visual Studio Code IDE. I used the JavaScript framework ReactJS, with HTML, CSS and Bootstrap to build the frontend. React allows me to create a single-page-application which will reduce the amount of page refreshes. The single-page-application will be composed of components.

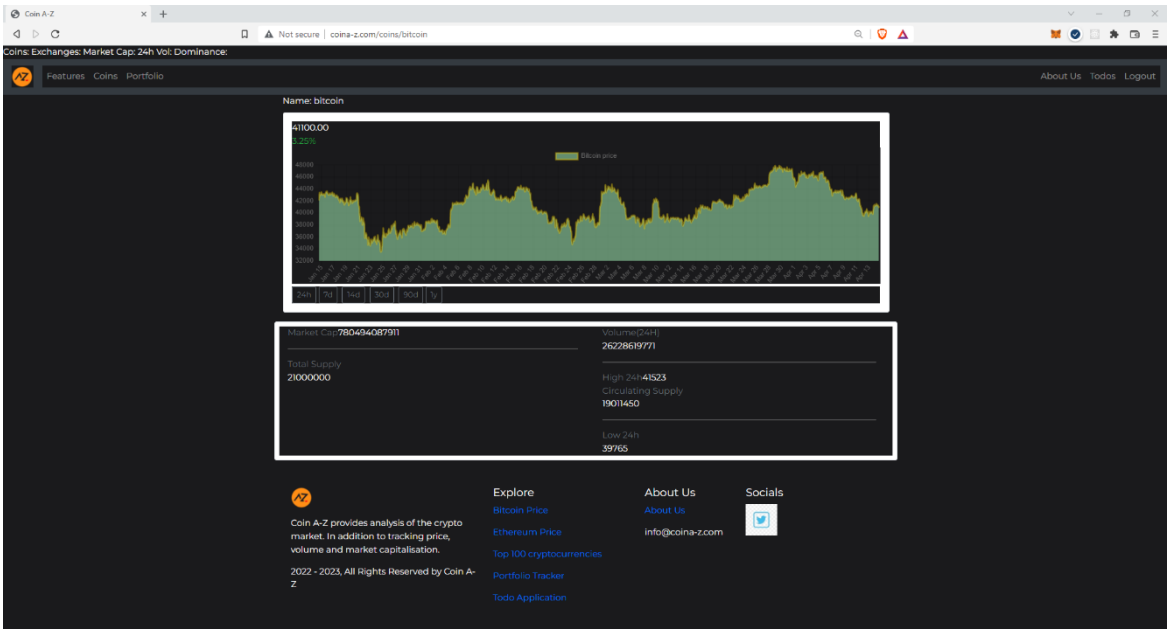
ReactJS

React (also known as **React.js** or **ReactJS**) is a free and open-source front-end JavaScript library for building user interfaces based on UI components. It is maintained by Meta (formerly Facebook) and a community of individual developers and companies. (ReactJS, 2022)

Libraries:

ChartJS:

> Chart.js is a free, open-source JavaScript library for data visualization, which supports eight chart types: bar, line, area, pie (doughnut), bubble, radar, polar, and scatter. (Chart.js, 2022)
>
> Using ChartJS I can display the coin data and graph it to make it an easy-to-read chart.



Bootstrap:

Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains HTML, CSS and JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components. (Bootstrap, 2022)

Axios:

Axios, which is a popular library is mainly used to send asynchronous HTTP requests to REST endpoints. This library is very useful to perform CRUD operations. (using-axios-for-api-call, 2022)

Using Axios allows me to share resources between the view and the controller. My services will call the REST API endpoints and manage the response from the controller.

Here is a sample of an Axios request, this is a simple request for just a ping with my backend whenever I was testing communication between my front end and my backend.

```
class portfolioDataService {
  retrievePing(name) {
    return axios.get(`${JPA_API_URL}/users/${name}/ping`);
  }
```

This is the file for App Component. App Component is the main component in React which acts as a container for all other components.

```
const App = () => {
  const [loginClicked, setLoginClicked] = useState(false);
  return (
    <div className="App">
      <Router>
        <>
          <HeaderDisplay />
          <NavigationBar loginClicked={loginClicked}  setLoginClicked={setLoginClicked}/>
          <Container>
            <Switch>
              <Route path="/" exact component={HomePage} />
              <Route path="/features" exact component={features} />
              <Route path="/coins" exact component={Coins} />
              <Route path="/coins/:id" exact component={CoinDetailPage} />
              <Route path="/login" exact render={(props) => <LoginComponent setLoginClicked={setLoginClicked} {...props} />} />
              <Route path="/register" exact component={RegisterComponent} />
              <Route path="/about" exact component={AboutPage} />
              <AuthenticatedRoute path="/welcome/:name" component={Welcome} />
              <AuthenticatedRoute path="/todos/:id" exact component={Todo} />
              <AuthenticatedRoute path="/todos" component={ListToDosComponents}/>
              <AuthenticatedRoute path="/portfolio" component={PortfolioList} />
              <AuthenticatedRoute exactpath="/logout" component={LogoutComponent} />
              <Route component={ErrorComponent} />
            </Switch>
            <Footer />
          </Container>
        </>
      </Router>
    </div>
  );
};
```

Backend

IntelliJ

IntelliJ IDEA is an integrated development environment written in Java for developing computer software. It is developed by JetBrains. (IntelliJ_IDEA, 2022)

IntelliJ is my chosen IDE for my backend development work, I have the most experience with using IntelliJ than any other IDE.

Java

Java is a high-level, class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible. It is a general-purpose programming language intended to let programmers *write once, run anywhere* (WORA), meaning that compiled Java code can run on all platforms that support Java without the need to recompile. (Java, 2022)

Spring Boot

Spring Boot is an open-source micro framework maintained by a company called Pivotal. It provides Java developers with a platform to get started with an auto configurable production-grade Spring application. With it, developers can get started quickly without losing time on preparing and configuring their Spring application. (what-is-spring-boot, 2022)

Spring frameworks provide inversion of control and dependency injections making CRUD operations and handling HTTP requests more straightforward.

The controller classes will act as a gateway and handle navigation. It will expose the endpoints and control access to the backend application by taking in user input and controlling the response.

Here is a sample of a Rest Contoller using Spring Boot, when a GET request is performed at this url it will return “ping success”. I was using this as a tester to make sure my comunication was working with my frontend and my backend, I used this as a basis for my other CRUD operations.

```
@RestController
@CrossOrigin(origins={"http://54.247.123.110/", "http://coina-z.com", "https://coina-z.com"})
public class AssetResource {
    @Autowired
    private AssetRepository assetRepository;

    @GetMapping("/jpa/users/{username}/ping")
    private String ping() { return "ping success!"; }
```

Here is my application properties file, I configure this file and specify my properties for the Spring Boot application.

application.properties

```
spring.datasource.url=jdbc:mysql://52.210.85.38/coinazuserdb?useSSL=false&allowPublicKeyRetrieval=true
spring.datasource.username=root
spring.datasource.password=password
spring.jpa.database-platform=org.hibernate.dialect.MySQL5Dialect
spring.jpa.hibernate.ddl-auto=create-drop
spring.jpa.hibernate.use-new-id-generator-mappings=false
spring.jpa.properties.hibernate.dialect = org.hibernate.dialect.MySQL5InnoDBDialect

logging.level.org.springframework = info
logging.level.org.hibernate.SQL= DEBUG

jwt.signing.key.secret=mySecret
jwt.get.token.uri=/authenticate
jwt.refresh.token.uri=/refresh
jwt.http.request.header=Authorization
jwt.token.expiration.in.seconds=604800
```

MySQL

MySQL is an open-source relational database management system. (MySQL, 2022)

This is the database I decided to use for my project as we have covered it in class and find it simple to use. I needed a SQL relational database as I plan to have structured data, where every record will have same properties and will ensure data integrity, especially where there are relationships between the data.

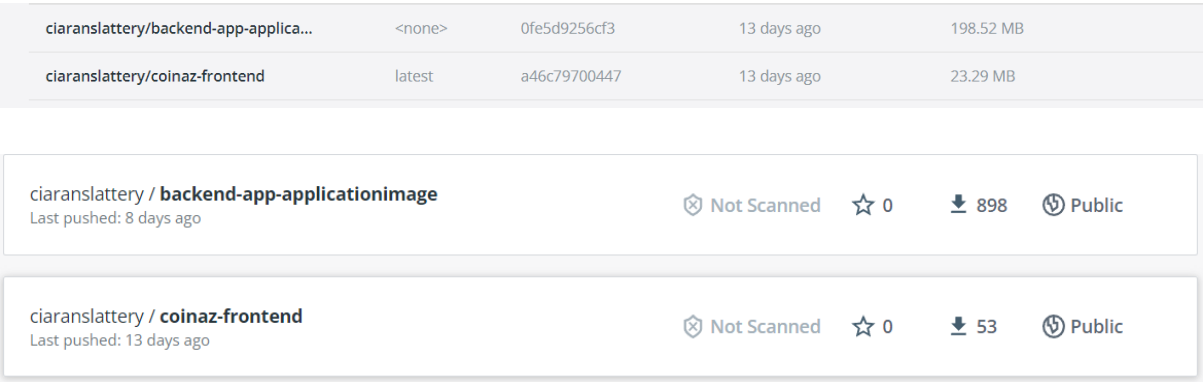
JWT (JSON Web Token)

JWT, or JSON Web Token, is **an open standard used to share security information between two parties — a client and a server**. Each JWT contains encoded JSON objects, including a set of claims. JWTs are signed using a cryptographic algorithm to ensure that the claims cannot be altered after the token is issued. (what-is-jwt, 2022)

Docker

Docker is an open source containerization platform. It enables developers to package applications into containers—standardized executable components combining application source code with the operating system (OS) libraries and dependencies required to run that code in any environment. Containers simplify delivery of distributed applications, and have become increasingly popular as organizations shift to cloud-native development and hybrid multicloud environments. (docker, 2022)

Using docker I wrapped my frontend and backend applications in docker images and used DockerHub to push my images onto. DockerHub is like GitHub in being a hosted repository service where I can push my docker images to and have easy access to them from anywhere.



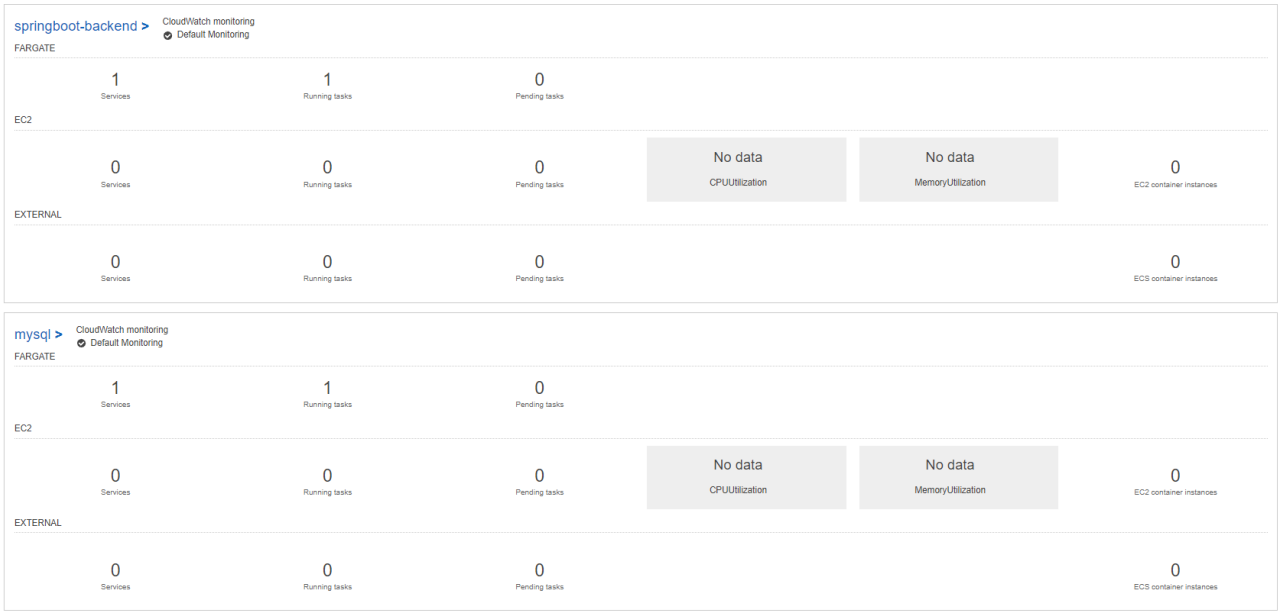
AWS (Amazon Web Services)

Amazon Web Services, Inc. (**AWS**) is a subsidiary of Amazon providing on-demand cloud computing platforms and APIs to individuals, companies, and governments, on a metered pay-as-you-go basis. These cloud computing web services provide a variety of basic abstract technical infrastructure and distributed computing building blocks and tools. One of these services is Amazon Elastic Compute Cloud (EC2), which allows users to have at their disposal a virtual cluster of computers, available all the time, through the Internet. (Amazon_Web_Services, 2022)

I used AWS ECS for hosting my application.

Amazon Elastic Container Service (ECS) is a highly scalable, high performance container management service that supports Docker containers and allows you to easily run applications on a managed cluster of Amazon Elastic Compute Cloud (Amazon EC2) instances. (ECS, 2022)

Using my DockerHub repositories I can pull my images straight into ECS and run the applications with ease.



7 Ethics

As web developers, we are responsible for shaping the experiences of user's online lives. By making choices that are ethical and user-centered, we create a better web for everyone.

1. Web applications should work for everyone
    - Built on top of progressive enhancement
    - Prioritize accessibility
    - Develop inclusive forms
    - Test with real users
2. Web applications should work everywhere
    - Built responsively
    - Value performance
    - Leverage off-line first capabilities
    - Expose permanent, human readable, deep links
3. Web applications should respect a user's privacy and security
    - Use https everywhere
    - Respect user tracking preferences
    - Provide users with clear information about how their information is used
    - Allow users to export their data
    - Secure user data
4. Web developers should be considerate of their peers
    - Comment and document code
    - Lint and test code
    - Make use of source control and continuous integration
    - Consume and contribute to open source when possible
    - Treat other developers with respect
    - Offer, follow, and enforce a code of conduct for open-source projects

(ethicalweb, 2022)

8 Conclusion

For my final year project, I set out to design a cryptocurrency tracking application and website using React, Spring Boot, MySQL, Docker, AWS. The website would allow a user to create an account, build a custom portfolio of coins and track their asset value. Users will also be able to look at coin data and graphs. I designed and developed from scratch a fully functional full-stack website which met the requirements I originally set out to complete. Using languages such as Java, JavaScript, HTML, CSS, Bootstrap and SQL allowed me to develop Coin A-Z, which allows users to easily understand and use the system without having ever used it before, with ease and efficiency. Using AWS, I hosted my frontend and backend applications, I also used DNS to host my website on a domain I acquired.

Having gone through the process of developing a software project from start to finish, I have enhanced my overall knowledge of each area SDLC (Software Design Life Cycle), which also incorporating and building on elements of modules that I have studied in my undergraduate course. The process has given me an understanding of the challenging work and planning that goes into software development projects and has allowed me to learn how to develop a full-stack application, which is valuable knowledge I will be taking with me further in my software career moving forward.

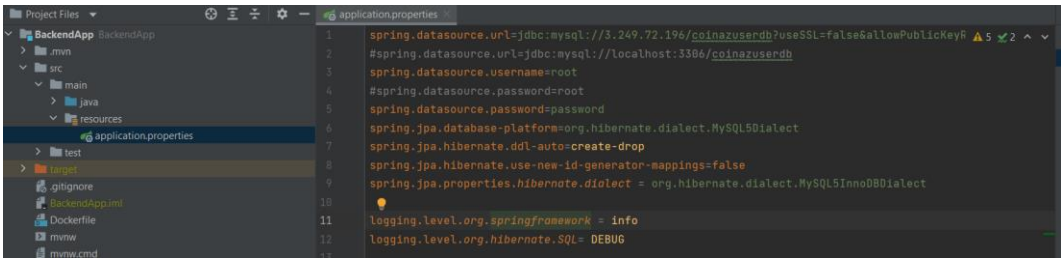
With further development of Coin A-Z I believe it can be a product with lots of potential and I plan to further develop this site as a hobby and to further enhance my understanding and skills.

9 Appendix

How to run Coin A-Z locally assuming MySQL running on port 3306:

1. Download a fresh repo: https://github.com/CiaranSlattery/CoinA-Z.git
2. Navigate to frontend and npm install
3. Navigate to Constants.js, update the URL in JPA_API_URL to 'http://localhost:8080/jpa'

4. You can now npm start the frontend directory and this should be all set.
5. Now navigate to the BackendApp.
6. Navigate to the applications.properties file.
7. Comment out the first line and uncomment the second line, this will now look for MySQL on your local machine. Uncomment the "spring.datasource.password= root" and comment out "spring.datasource.password= password".



8. Run the backend application and it should then connect to your local MySQL and the backend application will be hosted on localhost:8080.
9. Make sure your frontend and backend application are running and the website should have full functionality.

10 References

akana. (2022, April 16). *what-is-jwt*. Retrieved from akana.com: https://www.akana.com/blog/what-is-jwt

Amazon. (2022, April 17). *ECS*. Retrieved from aws.amazon.com: https://aws.amazon.com/ecs/faqs/

ansariaamir. (2022, April 16). *using-axios-for-api-call*. Retrieved from medium.com: https://ansariaamir.medium.com/using-axios-for-api-call-f63801926c74

Dawnie, N. (2022, April 16). *Chart.js*. Retrieved from wikipedia.org: https://en.wikipedia.org/wiki/Chart.js

https://en.wikipedia.org/wiki/React_(JavaScript_library). (n.d.).

https://www.geeksforgeeks.org/axios-in-react-a-guide-for-beginners/. (n.d.).

ibm. (2022, April 17). *docker*. Retrieved from ibm.com: https://www.ibm.com/in-en/cloud/learn/docker

Saputra, A. (2022, April 16). *what-is-spring-boot*. Retrieved from medium.com: https://medium.com/codestorm/what-is-spring-boot-2460fa6254e

Scott, A. (2022, April 19). *ethicalweb*. Retrieved from ethicalweb: https://www.ethicalweb.org/

wikipedia. (2022, April 15). *Amazon_Web_Services*. Retrieved from wikipedia.org: https://en.wikipedia.org/wiki/Amazon_Web_Services

wikipedia. (2022, April 16). *Bootstrap*. Retrieved from wikipedia.org: https://en.wikipedia.org/wiki/Bootstrap_(front-end_framework)

wikipedia. (2022, April 16). *GitHub*. Retrieved from wikipedia.org: https://en.wikipedia.org/wiki/GitHub

wikipedia. (2022, April 16). *IntelliJ_IDEA*. Retrieved from wikipedia.org: https://en.wikipedia.org/wiki/IntelliJ_IDEA

wikipedia. (2022, April 16). *Java*. Retrieved from wikipedia.org: https://en.wikipedia.org/wiki/Java_(programming_language)

wikipedia. (2022, April 16). *Jira*. Retrieved from wikipedia.org: https://en.wikipedia.org/wiki/

wikipedia. (2022, April 16). *MySQL*. Retrieved from wikipedia.com: https://en.wikipedia.org/wiki/MySQL

wikipedia. (2022, April 16). *ReactJS*. Retrieved from wikipedia.org: https://en.wikipedia.org/wiki/React_(JavaScript_library)