Ilyas Ismail . Jarvis Consulting

I'm a recent graduate with Honours from Centennial College with an Advanced Diploma in Software Engineering Technology. I am passionate about building applications with systems that will be utilized by many different users. I'm also an experienced Java and web developer who has built multiple Spring Boot applications, such as a library management tool for librarians to maintain their customers, books, and transactions in a MySQL database through CRUD features. I've also designed and developed ASP.NET and MERN-based web applications, such as a book store that allowed users to browse for books stored on a MongoDB database based on their search filter and add it to their cart for check-out. To summarize, I'm looking to put my experience and skills to good use in the software development industry.

Skills

Proficient: Java/Spring Boot, Node.js, Express.js, NoSQL, RDBMS/SQL, React, Amazon Web Services, Linux/Bash, Agile/Scrum, Git

Competent: C#, ASP.NET, Python, GraphQL, Jira, Azure Data Lake

Familiar: Angular, Apigee, Google Cloud Platform, Jenkins, PHP

Jarvis Projects

Project source code: https://github.com/jarviscanada/jarvis_data_eng_IlyasIsmail

Cluster Monitor [GitHub]: Implemented scripts and a database to track and store system specifications and resource usage to assist system administrators with maintaining system performance and faulty hardware. A PostgreSQL Docker container was created to host the database. Bash was used to retrieve the information needed through commands such as vmstat for memory, IO, and CPU usage statistics, df for available disk space, date for the current date in a specific format, and lscpu to retrieve the CPU's hardware details. The script responsible for collecting the system's resource usage was automated using a cron job that would run every minute. Testing was completed manually, running the script on a single machine, and debugging based on any errors thrown. Git and GitHub were used for version control using the GitFlow branching model.

Core Java Apps [GitHub]:

- JDBC App: Developed a Java 8 application to simulate purchasing and selling stocks based on real statistics from the Alpha Vantage REST API. The application takes one input from the user, the location of their properties.txt file. The application consists of three DAOs, two services, and one controller. The DAOs implement CRUD methods for stock quotes and positions to interact with a PostgreSQL database and the Alpha Vantage REST API. The services handle the business logic, such as ensuring the user is trying to purchase a valid amount of stocks, and calls on the DAOs to interact with the database. The controller acts as the console UI, it takes inputs from the user and passes them into the appropriate methods. Testing was done through unit, integration, and manual testing. Unit testing was done to ensure each method worked independently. Integration testing ensured the methods worked end-to-end, including retrieving and saving objects to and from the database. Manual testing was performed to confirm the console UI worked as intended, with SLF4J logs implemented to catch any errors. The application was deployed through Docker, with an image being created and published to Docker Hub for users to conveniently access. Git and GitHub were used for version control using the GitFlow branching method.
- Grep App: Developed a Java 8 application to mimic the Linux grep command. The application takes three inputs from the user: a regular expression pattern, the absolute path to the directory containing the files to be processed, and the absolute path for the output file produced by the application. The stream API is used within lambda expressions to retrieve all the files in the given directory and to store each line from the files into a list. Each line is then checked to see if it contains the pattern inputted by the user. Once all the files are processed, all matching lines are written and saved to the given output file path using BufferedWriter. Testing was completed manually, using different patterns and multiple files to check for errors through SLF4J logs. Deployment was done through Docker, a Dockerfile was created to build and upload an image to Docker Hub for users to consume easily. Git and GitHub were used for version control using the GitFlow branching method.

Highlighted Projects

Library Web and Mobile App [GitHub]: Collaborated with a team of four to develop a cloud-hosted web application and an accompanying mobile application over four months. Customers could search for books based on filters such as genre or title and were able to add books to their shopping cart. Admins were able to apply CRUD functions on books

for maintenance. We implemented the app using the MERN stack alongside libraries such as Axios, JWT, and bcrypt for connection requests and authentication, Bootstrap for better front-end visualization, and Nodemailer to send e-mails. We hosted the web app online through Heroku and the mobile app through Google Firebase. The mobile app made requests directly to and from the web app's API. Testing was done through manual testing of each feature.

Video File Sharing Site: Developed an MvC ASP.NET web application that allowed users to register, log in, and upload or edit their video files for other users to download and comment on to leave a review. The uploader would provide the title, genre, description, and release date of the video. The other two columns, uploader, and ratings, are automatically generated based on other values such as the uploader's account name and the comments provided. Other users can comment on videos and leave a rating out of ten, the average rating of the video is the video's overall rating. DynamoDB was used to store all the text values for every movie object. Amazon S3 was used to store and download the video files. Elastic Beanstalk was used to host the application online.

Library Management Tool: Designed and developed a Spring Boot Java application that allowed librarians to apply CRUD functionalities to customers, books, and transactions. All controllers had CRUD features implemented, the books controller also allowed the librarian to view which books were available and which books were loaned out. Maven was used for build automation, to get all the required packages. MySQL was used to store the data for all three models. Apache Tomcat was used to locally host the application.

Professional Experiences

Software Developer, Jarvis (2024-present): Developed programs in Java and Linux/Bash using a Docker PostgreSQL container as the database, and Git for version control. Supplied each project with a detailed documentation to help others navigate and understand each project developed. Practiced Agile methodology through daily scrum meetings, sprint planning, and sprint retrospectives to ensure the best quality of work.

Education

Centennial College (2020-2024), Advanced Diploma in Software Engineering Technology, Engineering Technology and Applied Science - Graduated with Honours

Miscellaneous

- I enjoy playing basketball and volleyball, I played for school teams and my local league
- I enjoy playing strategic team-focused games such as League of Legends, Rainbow Six Siege, and Counter-Strike 2
- I recently started learning Blender to create 3-D models.