**PANCAKES**

**F U L L S T A C K D E V**

# APDS7311 **PART 2**

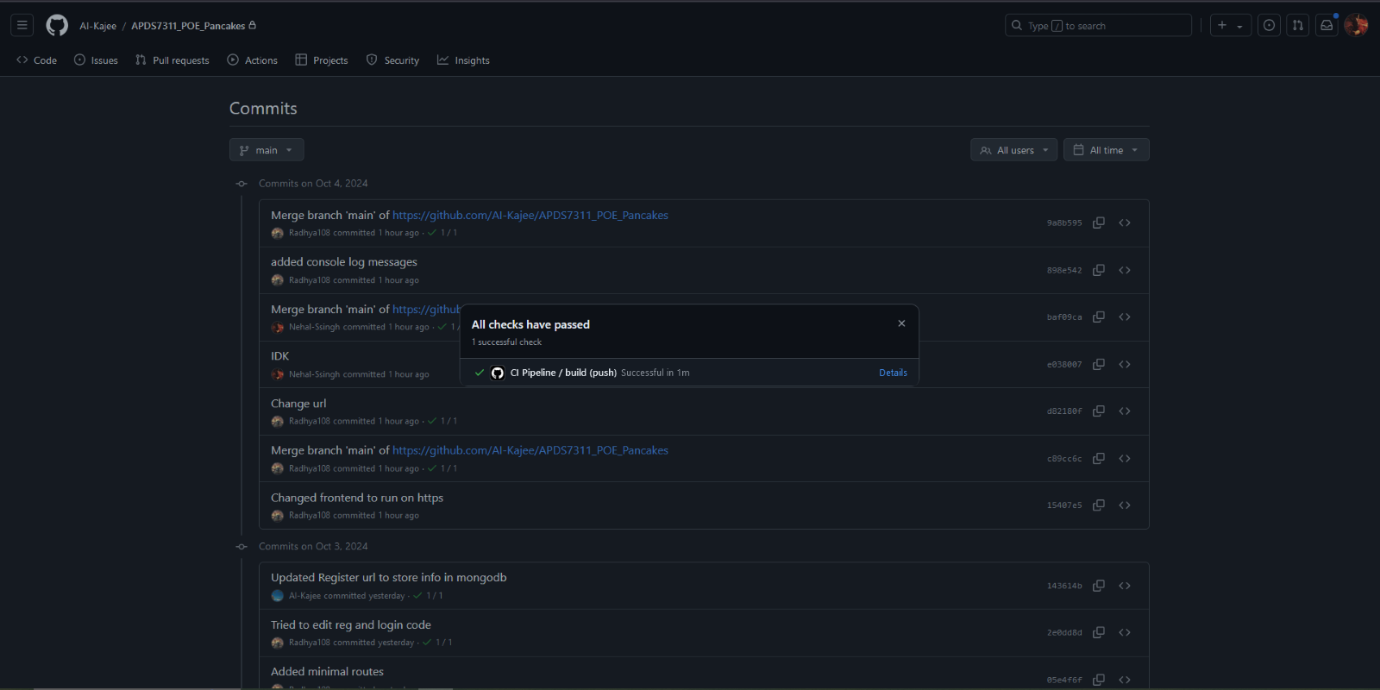
This document serves as proof for our additional research.

**Ameer Inder Kajee - ST10063110**

**Nehal Singh - ST10184628**

**Radhya Chinnapen - ST10069610**

Github Actions Pipeline for DevSecOps



**Overview:**

The process of automatically merging code changes into a designated branch of a repository is known as continuous integration, or CI. It greatly reduces the time needed for manual testing by executing tests and creating artifacts to guarantee code quality and functionality.

Code updates can be deployed whenever they pass tests thanks to Continuous Delivery (CD), which expands on Continuous Integration (CI) by automating the release process. Deploying staging environments and running additional tests are part of this step.

By automatically deploying verified changes to the production environment without requiring human intervention—if predetermined conditions are met—Continuous Deployment goes one step further. Strict CI procedures are needed at this point to guarantee deployment safety.

A solution for integrated automation called GitHub Actions makes the CI/CD process easier within GitHub projects. YAML files can be used by developers to build workflows that automate processes such as code merging, testing, and packaging artifacts. In addition to managing deployments, notifications, and other automation tasks, GitHub Actions facilitates third-party integrations. (Kerno, 2024).

# **Key Benefits:**

Automation: Lessens the need for manual intervention, enabling more dependable and frequent releases.

Efficiency: By automating testing and deployment procedures, it shortens the development lifecycle.

Integration: Easily connects to third-party services and current GitHub repositories. (Kerno, 2024).

Utilizing continuous integration and continuous delivery (CI/CD) to automate critical activities is crucial to modern DevOps methods. One particularly effective solution for CI/CD pipeline implementation is GitHub Actions, which offers a streamlined and effective method for software development and deployment. Teams may expedite their development lifecycle and maintain high code quality by employing GitHub Actions, freeing up developers to concentrate on coding while automating builds, tests, and deployments. (Kerno, 2024).

<https://youtu.be/-FJ6O3bWa0E>

<img src="https://github.com/ST10055763/HealthCents/blob/master/images/R.jpeg" alt="Heroku" width="300">

Protecting Against Attacks

### **SSL Certificates**

In our code we made use of SSL Certificated in both our frontend and backend to encrypt data in transit and mitigate Man-in-the-Middle attacks.

An SSL Certificate guarantees safe data transmission between a visitor and a website, protecting against Man-in-the-Middle (MitM) attacks, when a hacker intercepts communication between the client and server.

Exploiting unencrypted data delivered over vulnerable protocols like HTTP, MitM attacks are common. SSL certificates use asymmetric cryptography to encrypt data and validate identities in order to upgrade connections to HTTPS. Because they do not have the private key, hackers are unable to decrypt information that has been intercepted. (Munteanu, 2019).

### **Prevention Against SQL injection and Session Jacking**

Bcrypt protects against session jacking and SQL injection, we relied on Bcrypt in our project to protect session tokens and to hash passwords and other sensitive information before storing them in mongoDB. (Heller, 2020).

For additional security we implemented crypto to handle the encryption and decryption of the sensitive payment information. It offered flexibility as it allowed us to decrypt the information to display the payment history to the user, whilst maintaining its security. Crypto utilises a key, which we determine, to create a cipher for encrypting and decrypting the information. We ensure the key is secured by creating the environment file (.env) and storing both the encryption key and mongo url. (Data Encrypt — Decrypt in React Application, 2023)

Mongo-Sanitize has been used to sanitize user inputs before interacting with them in the database, this prevents malicious inputs from users, catering towards preventing SQL injection. The middleware package is called Express Mongo Sanitize and it is made to sanitize user input, especially to guard against NoSQL injection attacks in Node.js apps that use MongoDB. This middleware ensures that your application is safe from potential NoSQL injection vulnerabilities by automatically sanitizing any user input submitted in the query parameters or request body. (Işik, 2020).

Another action that we took in countering session jacking was session management, we implemented secure session management with HttpOnly and secure cookie flags. (Kumar, 2023).

### **Cross-Site Scripting and Click jacking**

xss-clean is a middlewear that we use to sanitize user input and prevent any XSS attacks,

**Functionality**:

* **Sanitizes inputs from**:
* **POST request body** (req.body)
* **GET request query** (req.query)
* **URL parameters** (req.params)

This middleware isessential for maintaining the security and integrity of your web application by mitigating the risks associated with user-generated content. (Rathi, 2021).

We implemented the Helmet middleware to enhance security by setting various HTTP headers, including X-Frame-Options and Content Security Policy, in our HTTPS-enabled application.

**Automatic Header Configuration**: Helmet automatically configures important HTTP headers like X-Content-Type-Options, X-Frame-Options, and Strict-Transport-Security, reducing the need for manual configuration.

**XSS Protection**: Helmet enables the X-XSS-Protection header, which can mitigate some XSS attacks, particularly in older browsers. (GeeksforGeeks, 2024).

### **DDoS**

To counteract DDoS Attacks we applied rate limiting with express-rate-limit to control the number of requests from a single IP address.

windowMs: Specifies the millisecond period in which the rate limit is in effect. For instance, a 15-minute window is set with windowMs: 15 \* 60 \* 1000.

 max: Indicates the maximum quantity of requests that are permitted within the time span to come from a single IP address. The rate limit will be activated in the event that requests above this predetermined limit. (Manning, 2021).

Design Process

Atlas Trust is designed to provide banking clients with a secure and efficient platform for managing their international transactions. We opted for a more user-centric approach, to make their experience more enjoyable and simplistic.

Our team analysed other banking platforms such as Investec, Wise and Retail Capital.

Click below to visit these sites:

* **Investec:** <https://www.investec.com/en_za.html#:~:text=Login%20to%20your%20Investec%20accounts.%20Secure%20login%20to%20Investec>
* **Wise:** <https://wise.com/>
* **Retail Capital:** <https://www.retailcapital.co.za/?msclkid=4ed15bf87fc010f1e0913cfb879ef3b2&utm_source=bing&utm_medium=cpc&utm_campaign=Online_BA_PrimaryFocus_CONV_Mar%2723_FA_New_SA__GSN_&utm_term=business%20funding&utm_content=%5BDMT%5D%5BDSA%5D%20Small%20Business%20Funding>

We prioritised the needs of our target audience:  
**Audience:** Clients banking with Atlas Trust

**Needs and Preferences:**

|  |  |  |  |
| --- | --- | --- | --- |
| Client Needs | Description | Research | Our Implementat |
| Intuitive Navigation | Users need a clear and straightforward navigation structure to easily find features and services. | All three sites made use of hyperlinks as to not overcrowd the space with buttons and card views to combine helpful messages with direction. Each had a navigation bar | We included a navigation bar with hyperlinks to direct them to their available functions. And a dashboard that is a centralised point for users to refer to. |
| Quick Access to Transactions | Clients want to view their recent transactions quickly without navigating through multiple pages. | Investec has a tab area that categories the transactions. Click on any of the tabs to go to the ordered lists. | After each payment they are directed to their list of transactions with clear headings making it easy to understand. |
| Clear Call-to-Action Buttons | Clients need prominent, easy-to-understand buttons for completing actions like payments or transfers. | Wise implements clear buttons that stand out against the white background. It keeps the buttons in clear areas and avoids overloading the user with too many. | We chose consistency with our button colours and sizes, layout of information and messages. |
| Visual Feedback on Actions | Clear messages to direct them in their process. | Retail Capital makes use of clear descriptions and diagrams such as the steps involved for transactions. | Error messages were boldly displayed in a red box with icons and confirm messages were bright with a clear font. |
| Simplicity | Users required consistency to understand where functions are, what they do and not distract them from the core of the website | Wise sections out content and functionality a lot better. They are not overly descriptive. The site avoids putting too much onto one screen, adding more basic navigation and design consistency. | We kept our font type to Nunito, maintained the same font sizes for specific areas such as headings and messages. The colour scheme: we stuck to a specific palette of blues, greys and white. |

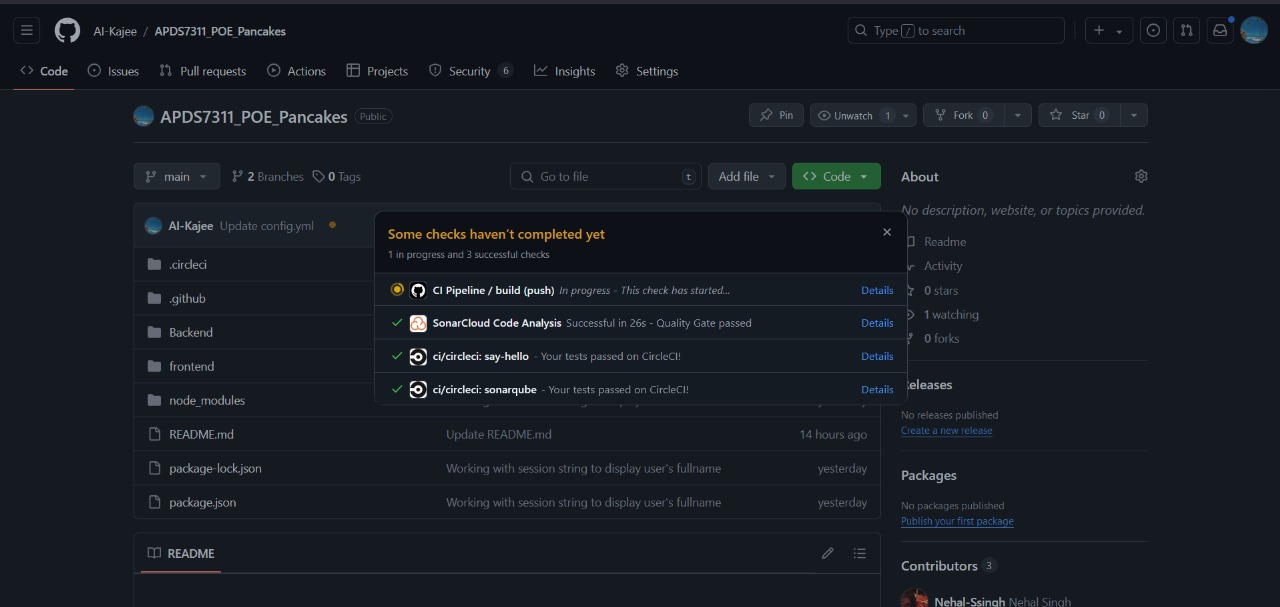
These websites were brilliant examples of user focused design. However, we still wanted to add our personal touch so we utilised two tools to achieve this:

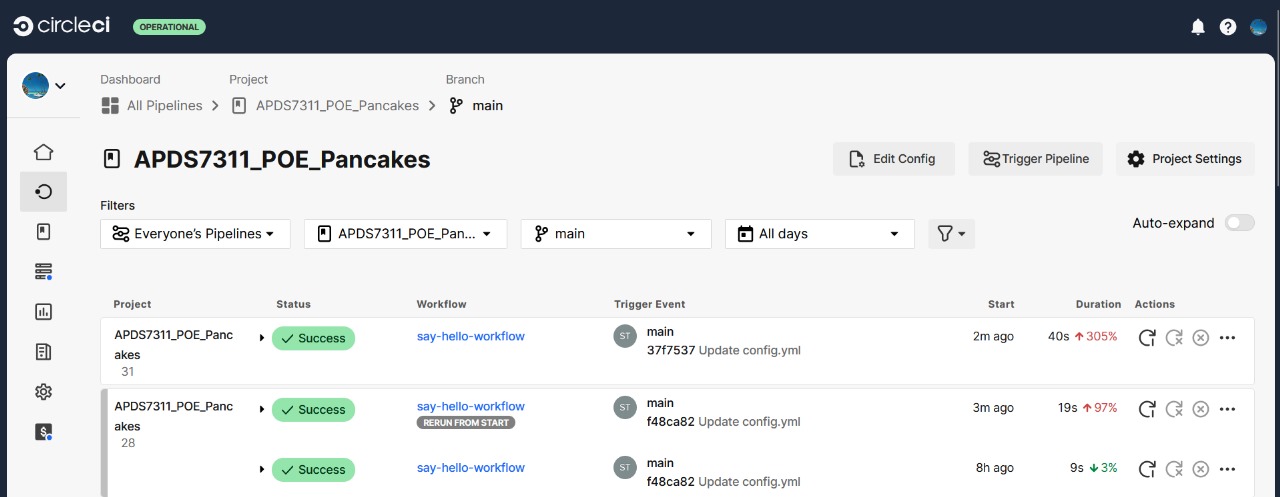
**Figma:**   
Implemented for wireframes, allowing us to map out our functionalities and user navigation. It helped us refine the navigation to make sure we could pass information appropriately between screens and make user navigation as seamless as possible. It also ensured we maintained that design consistency throughout. It was a great tool for brainstorming ideas as a team.

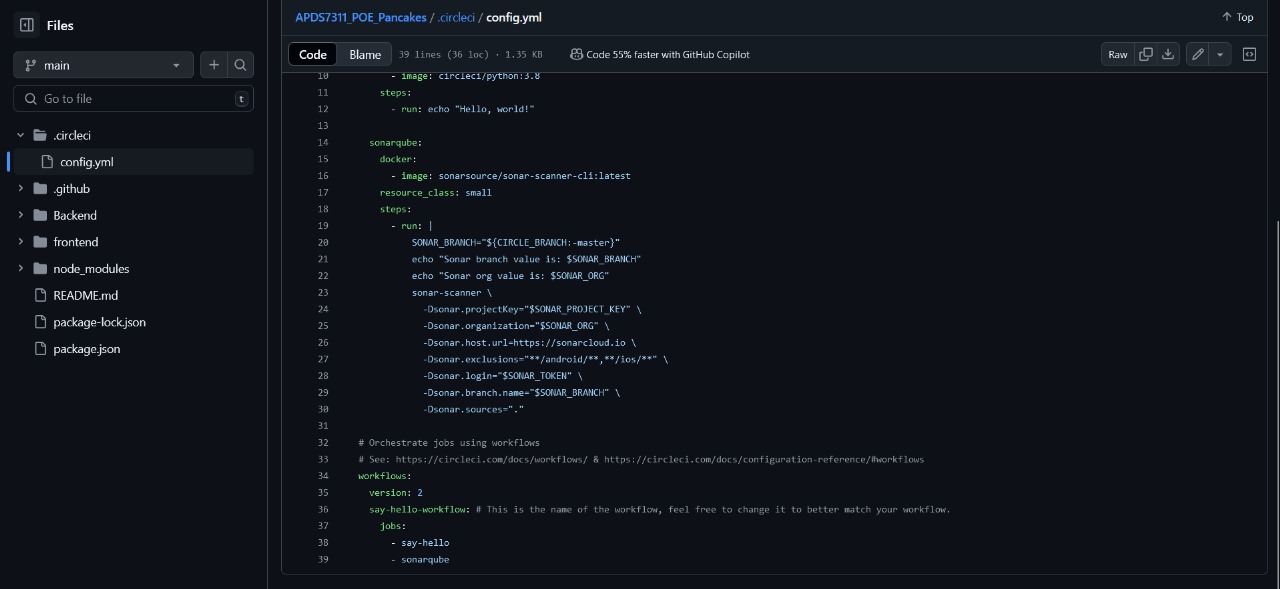
##### **Canva:**

We took inspiration for our company name from the purpose of the app. Atlas was chosen as we focus on international payments. We used Canva to create a colour palette, design our logo and background images.

Circle CI – Sonar qube Pipelines







Reference List:

Data Encrypt — Decrypt in React Application (2023) DZone. Available at: https://dzone.com/articles/data-encrypt-decrypt-in-react-application (Accessed: 8 October 2024).

Kerno. (2024). *CI/CD pipelines with GitHub Actions*. Available at: <https://www.kerno.io/learn/ci-cd-pipelines-with-github-actions> (Accessed: 5 October 2024).

Munteanu, A. (2019) ‘SSL Certificates vs. Man-in-the-Middle attacks’, *Medium*. Available at: <https://medium.com/@munteanu210/ssl-certificates-vs-man-in-the-middle-attacks-3fb7846fa5db> (Accessed: 7 October 2024).

Heller, A. (2020) ‘Storing user passwords with bcrypt’, *Medium*. Available at: <https://medium.com/@andrewhheller/storing-user-passwords-with-bcrypt-e3b7306b870a> (Accessed: 7 October 2024).

Işik, H. (2020) ‘Mongo NoSQL injection attack and how to prevent them with Node.js: Express code examples’, *Medium*. Available at: <https://medium.com/@huseyin.isik000/mongo-nosql-injection-attack-and-how-to-prevent-them-with-nodejs-express-code-examples-beebae1a3d98> (Accessed: 7 October 2024).

Kumar, D. (2023) ‘HTTP-Only session hijacking through XSS’, *Securium Solutions*. Available at: <https://securiumsolutions.com/http-only-session-hijacking-through-xss> (Accessed: 7 October 2024).

Rathi, N. (2021) '5 NPM Packages to Secure Your Node.js Backend in 5 Minutes', *DEV Community*. Available at: <https://dev.to/itsnitinr/5-npm-packages-to-secure-your-node-js-backend-in-5-minutes-2732> (Accessed: 7 October 2024).

GeeksforGeeks (2024) 'Explain the purpose of the Helmet middleware in Express JS', *GeeksforGeeks*. Available at: <https://www.geeksforgeeks.org/explain-the-purpose-of-the-helmet-middleware-in-express-js/> (Accessed: 7 October 2024).

Manning, D. (2021) 'Securing APIs: Express rate limit and slow down', *Mozilla Developer Network*. Available at: <https://developer.mozilla.org/en-US/blog/securing-apis-express-rate-limit-and-slow-down/> (Accessed: 7 October 2024).