**Corbin Peever – CC Assignment 3 – Solution Architecture**

**Links**

**Project URL:** <http://ccassignment3.ap-southeast-2.elasticbeanstalk.com/>

**Project API Invoke URL:** <https://q783t786z8.execute-api.ap-southeast-2.amazonaws.com/prod> OPTIONS= /users, /posts, /likes, /friends, /comments

**GitHub:** <https://github.com/CorbinCoder/cc-assignment-3.git>

**Summary**

My Project, LookBook, is a social-media website that allows users to register an account with display pictures, log in, create posts with images, comment on posts, like posts, view other user profiles, add other users as friends, remove them as friends, and search for users or posts. It includes a RESTful API hosted on AWS Lambda, which uses AWS API Gateway to facilitate requests. Data is stored in a DynamoDB NoSQL database, image files are stored on S3, and session data is stored in RDS. CloudFront is used to quickly serve images from the S3 bucket. The application is written in Python, using Flask and Werkzeug for WSGI functions, Jinja for HTML logic, boto3 to interact with AWS services, and MySQL for database operations.

**Introduction**

1. **Motivation:** My motivation behind creating a social media website was that I wanted to gain an understanding of the fundamental requirements of making a web application of this type. I am interested in products of this type and would like to develop a more fully-fledged version in the future.
2. **High-level view:** At a high level, LookBook acts as a social media website that allows users to interact with one another socially. They can create their own profile, add friends, and engage with other users by creating, commenting on, or liking posts
3. **Beneficiaries:** The main beneficiaries of this application are the users that would interact with it, as it would bolster their ability to engage with other people online. The other clear beneficiaries are the owners of the product, currently me, if/when it is eventually monetized.

**Related work**

**Facebook:** <https://www.facebook.com/>

**X:** [https://x.com/](https://x.com/?lang=en-au)

**Instagram:** <https://www.instagram.com/>

**System Architecture**

**Compute:** EC2 using Elastic Beanstalk, Lambda

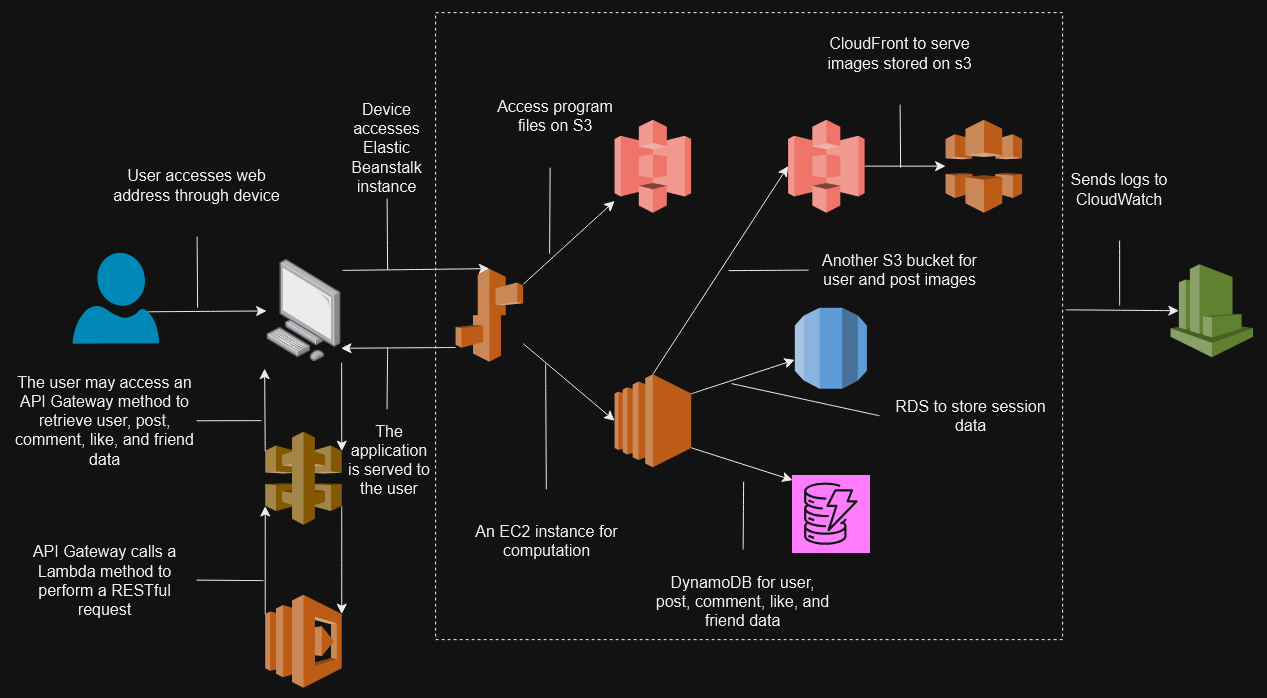
**Containers:** Elastic Beanstalk

**Storage:** S3

**Database:** DynamoDB, RDS

**Analytics:** CloudWatch

**Networking and Content Delivery:** API Gateway, CloudFront



*Figure 1.0 – Cloud Services Architecture*

**References**

* raucci2000 (2019) *The session is unavailable because no secret key was set. · issue #107 · Pallets-ECO/Flask-session*, *GitHub*. Available at: https://github.com/pallets-eco/flask-session/issues/107 (Accessed: 19 November 2024).
* Maurya, A. (2024) *Build a serverless web application with Aws Lambda ...*, *Building a Serverless Web Application with AWS Lambda, API Gateway, DynamoDB, S3*. Available at: https://aws.amazon.com/getting-started/hands-on/build-serverless-web-app-lambda-apigateway-s3-dynamodb-cognito/ (Accessed: 21 November 2024).