

## National College of Ireland

### Project Submission Sheet

**Student Name:** Anurag Singh  
**Student ID:** X23180013  
**Programme:** Master of Science in Cloud Computing **Year:** Jan 2024-2025  
**Module:** Cloud Platform Programming  
**Lecturer:** Shivani Jaswal  
**Submission Due Date:** Tuesday 23th April 2024, 4:00 pm  
**Project Title:** RoundLab: Beauty retail using AWS  
**Word Count:** 3604

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

**ALL** internet material must be referenced in the references section. Students are encouraged to use the Harvard Referencing Standard supplied by the Library. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action. Students may be required to undergo a viva (oral examination) if there is suspicion about the validity of their submitted work.

**Signature:** Anurag Singh  
**Date:** 23/04/24

#### PLEASE READ THE FOLLOWING INSTRUCTIONS:

1. Please attach a completed copy of this sheet to each project (including multiple copies).
2. Projects should be submitted to your Programme Coordinator.
3. **You must ensure that you retain a HARD COPY of ALL projects**, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer. Please do not bind projects or place in covers unless specifically requested.
4. You must ensure that all projects are submitted to your Programme Coordinator on or before the required submission date. **Late submissions will incur penalties.**
5. All projects must be submitted and passed in order to successfully complete the year. **Any project/assignment not submitted will be marked as a fail.**

#### Office Use Only

Signature:

Date:

Penalty Applied (if applicable):

# RoundLab: Beauty Retail using AWS

Anurag Singh

StudentId: x23180013

Cloud Platform Programming, National College of Ireland Dublin, IRELAND

Email: x23180013@student.ncirl.ie. URL: www.ncirl.ie

Deployed Application URL: <http://x23180013-cpp-project-env.eba-vbitjavb.eu-north-1.elasticbeanstalk.com>

Python Library link: <https://pypi.org/project/search-view-pkg-Anurag09/>

**Abstract**—Cloud computing has transformed the field of the software industry by providing easy access to computer resources and overcoming traditional infrastructure limitations. By leveraging this technology, RoundLab is a web application that successfully streamlines e-commerce operations such as viewing available products, placing orders, completing payments and tracking orders for customers. The project provides a comprehensive outline of the different components, their architectural design, continuous integration, and deployment processes, the creation library for reusable components and developmental insights that went into the making. The application is developed using Python and Django framework on Cloud9. The CI/CD pipeline, built using Amazon Web Services (AWS) components such as Elastic Beanstalk, CodeBuild, CloudWatch and CodeDeploy, has significantly automated and orchestrated the development process. The pipeline deploys code changes seamlessly, making it easy and efficient to deploy new features. Developers can make code changes and push them to a code repository, which then triggers the pipeline to build, test, and deploy the code. This process ensures that the code is always up-to-date and available to customers. This study showcases one aspect of how this technology can address challenges in the retail industry. The report provides details on all the development stages of the application. By integrating all of these cloud services, we can gain a better understanding of how they reduce costs, time, and resources while increasing availability and scalability. This will help us to have a clearer picture of the benefits of cloud computing.

**Index Terms**—Cloud computing, Python, Django, Continuous Integration and Continuous Deployment (CI/CD), Cloud9, Elastic Beanstalk, CodeDeploy, CodeBuild, CloudWatch.

## I. INTRODUCTION

The rise of cloud computing has simplified the development, deployment, and management of software applications, providing novel opportunities for innovation and collaboration in the software engineering process [1]. It has emerged as a game-changer in software engineering and development. RoundLab is a platform that uses cloud-based technology to help transform the online shopping booking experience for beauty products making it more personalised, supporting organic manufacturing, seamless distribution and listening to adapt to customer feedback. It integrates cloud computing technologies using AWS, Django framework as a backend for API creation and templates using HTML5, CSS3 as a frontend for rendering views and DevSecOps ideas to boost operational efficiency, reduce offline buying, auto-scaling retail infrastructure, and expedite the ordering process. The application is developed in two phases: the execution of essential

functionalities, as defined in the functional requirements, and the incorporation of AWS services, as detailed in the non-functional requirements.

### • Functional Requirements

- 1) User Authentication: For user authentication, the application supports login functionality for already registered users and a signup page for new users.
- 2) User Roles: The application supports two types of roles Admin and Customer.
- 3) Add and Browse products: Admins can add, modify, view and delete products whereas customers can view and add products to a cart.
- 4) Track order: Admins can manage the orders by updating the status such as Pending, Order Confirmed, Out for Delivery and Delivered, Cancel the order by deleting it. Customers can track the status of their orders additionally they can download invoices of their orders.
- 5) Customer profile: On signup, customers create their profile and also update it later. Admins can view, edit and delete customer profiles.

### • Non-Functional Requirements

- 1) Availability: As the application is deployed in AWS Elastic Beanstalk, it helps in maintaining zero downtime.
- 2) Usability: As the workflow in the application UI smooth and simple, allowing customers to easily order and browse with interactive elements in UI.
- 3) Scalability: As CodeBuild and AWS Elastic Beanstalk can auto scale on demand to increase performance at reduced cost. Reusability: The creation of python library helps in reusability of the developed code which maintaining high coding standards.
- 4) Performance: The performance is increased is on-demand scalability and almost zero delay in API response. AWS Elastic Beanstalk assures dynamic scalability without affecting performance.
- 5) Security: The application has authentication and authorisation which are implemented using Django middleware.

## II. ARCHITECTURAL DESIGN

This section as outlined in the figure 1 explains the architecture design of RoundLab.

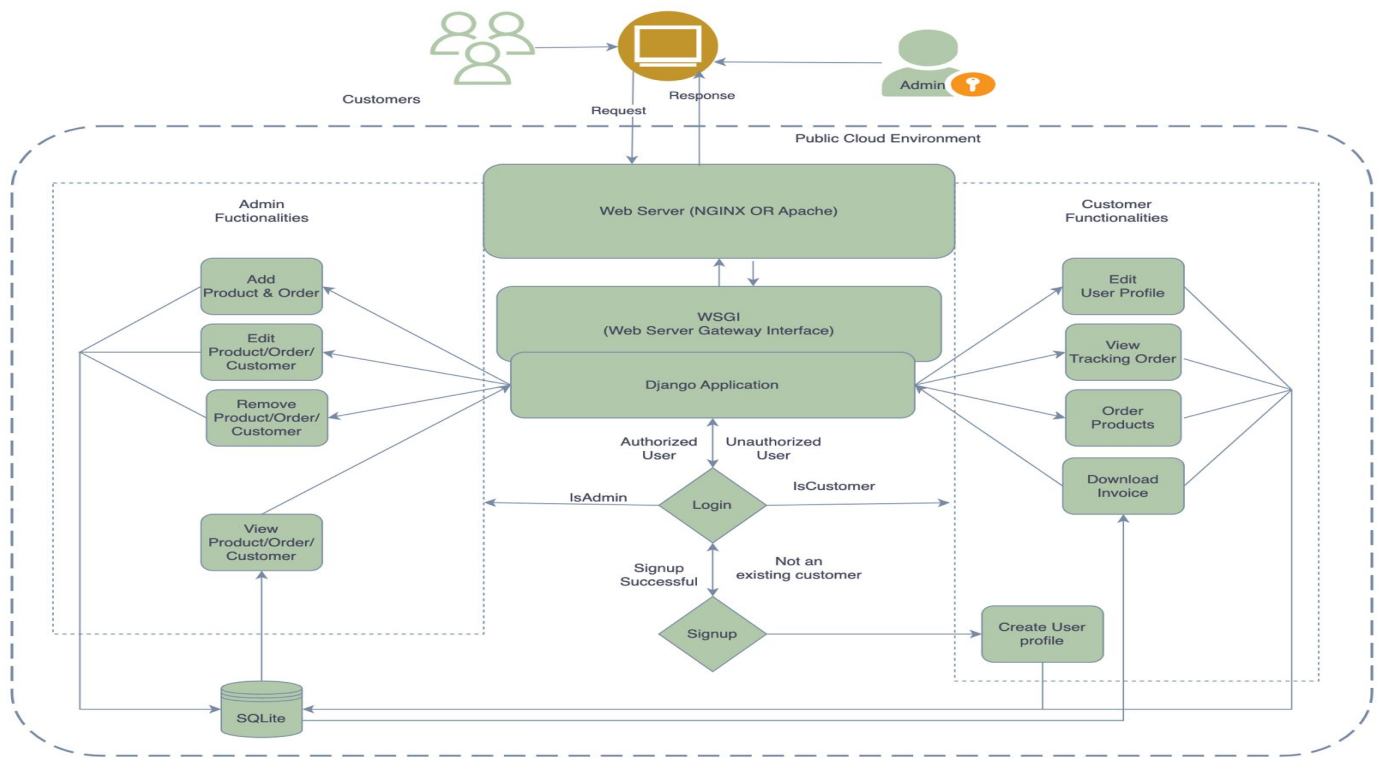


Fig. 1. RoundLab architecture diagram

The application supports 2 types of users Admin and customers. Both the user roles access the application hosted on AWS Elastic Beanstalk. The code is developed in Python and Django framework. The architecture of the Django framework follows a Model/View/Controller design pattern providing compatible, scalable, and maintainable applications. While the front end is made using HTML5 and CSS3 to render user-friendly views. This made the application interactive providing a seamless flow for shopping online. The application avoids the use of any third-party database but SQLite which comes integrated with the Django framework.

The application supports multiple features for both user roles Admin and customer or user. The Admin can log in using the admin login portal, which is redirected to the admin dashboard after successful authentication and authorization. The dashboard consists of a list of recent orders the customers gave along with customers, orders and product collection count. The admin navbar contains navigation to customers and products management pages where the admin can add, delete or modify any product but the admin can only delete or modify a customer's profile. Also, there is a view order link in the admin navbar which allows the admin to view all the orders along with the privilege of updating the status such as Pending, Order Confirmed, Out for Delivery and Delivered, Cancel the order by deleting it.

The customer functionalities include registering into the application for the first time which will help to create their profile. Once registration is completed they will be asked to

log in to the application. After that, the customers can browse through the various products, add selected products to the cart and place an order by playing through the payment portal. Customers can also track their orders and download an invoice for the order placed. They can also edit their profile which was created during the registration process.

From a developer's perspective the architecture of the application is an encapsulation of a full-stack application using the Django framework, HTML templates and SQLite database in a public cloud environment which in this case is AWS as shown in the figure 2 . The application code development and workflow design are done on AWS Cloud9. It is an integrated development environment that you can access through your browser online for writing, running, and debugging code. Once the code is developed, GitHub is used for version controlling of the application where the code is committed and tracking the changes in the code-base for better code maintenance.

CodeBuild is used to test and build the application. The build spec file for code build includes commands for installing and running pylint for static analysis techniques to discover code quality concerns and security vulnerabilities, ensuring the application codebase is resilient and secure. CodeDeploy is used to deploy the application code to instances automatically to enable fast and reliable application upgrades. Simple Storage Service (S3) is used in the application for storing all the static files and receipts of invoices created by the user after completing the order workflow.S3 provides features such as storage optimization, organizing and access management of

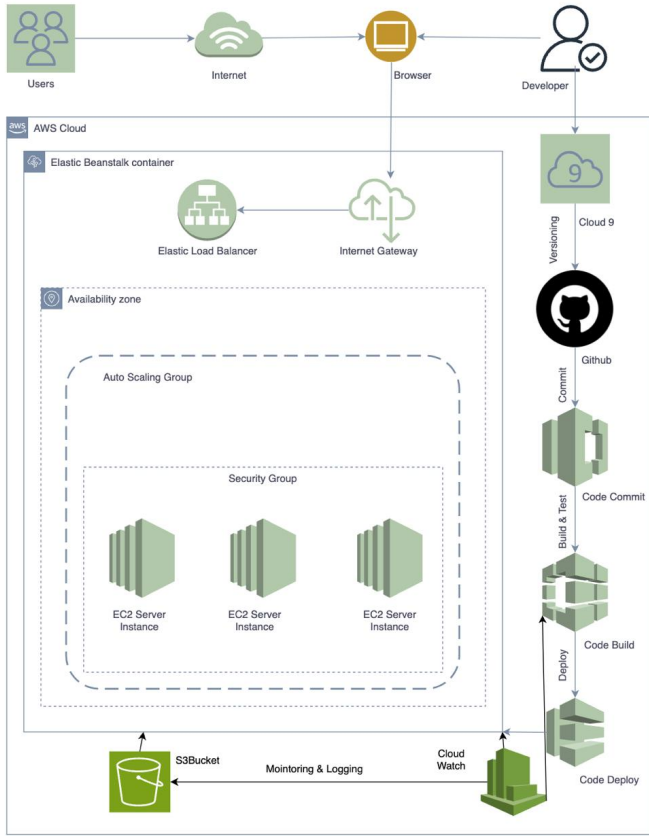


Fig. 2. AWS cloud architecture

the data stored along with monitoring and logging of the data through CloudWatch or CloudTrail.

The application is deployed in AWS Elastic Beanstalk [2] which can swiftly install and maintain apps on the AWS Cloud without knowing or worrying about the infrastructure that supports them. Elastic Beanstalk decreases administration complexity without limiting options or control by uploading the application, and Elastic Beanstalk will handle capacity provisioning, load balancing and scaling. It automatically starts an environment and sets up the AWS resources required to run the code. The Elastic Beanstalk console, or eb, is a high-level command-line interface created exclusively for Elastic Beanstalk. It automatically starts an environment and sets up the AWS resources required to run the code.

For logging, monitoring and tracking of AWS resources like S3, CodeBuild and Elastic BeanStalk, CloudWatch is used in the application architecture to track the health of the above resources, setting alerts for billing or other metrics to be notified if the threshold set is crossed. I have also created CloudWatch dashboard by setting up all the necessary widgets required for the monitoring or logging of the cloud resources along with setting alerts for any breakdown or billing updates. CodePipeline is used for orchestrating continuous integration and continuous deployment(CI/CD) pipelines by automating

the release process from code commit to deployment.

### III. AMAZON WEB SERVICES

Amazon Web Services has more than 200 services which include simple storage to fully-fledged virtualized infrastructure. With AWS features, web applications may seem to be as easy as a matter of minutes, without the need to provision servers on top of it, plus additional lines of code. While I think that there are so many services which can occur in this, having experimented on this project, we employed the simplest and most fundamental of the services. The services which are used in my application are as follows:

#### A. Amazon Elastic Beanstalk

Elastic Beanstalk is a service provided by Amazon that simplifies deployment and manages applications on the AWS cloud. There are two main functionalities of Elastic Beanstalk Auto-Scaling and Load Balancing. To deploy an application using Elastic Beanstalk, I created a local application using Python. Then, created an application environment in Elastic Beanstalk where I pushed the local application through the pipeline 3. Elastic Beanstalk then automatically calculates and provides the number of EC2 instances required along with S3 buckets. The application can be stored in different versions. Elastic Beanstalk allows the storage of different versions of the applications by storing the application versions on S3 buckets. Additionally, CloudWatch triggers an alarm for scaling if the average outbound traffic of every instance is more than 6 Mib and if it gets lower than 2 Mib in five minutes. Along with monitoring environment health, request count, target response time and CPU utilization.

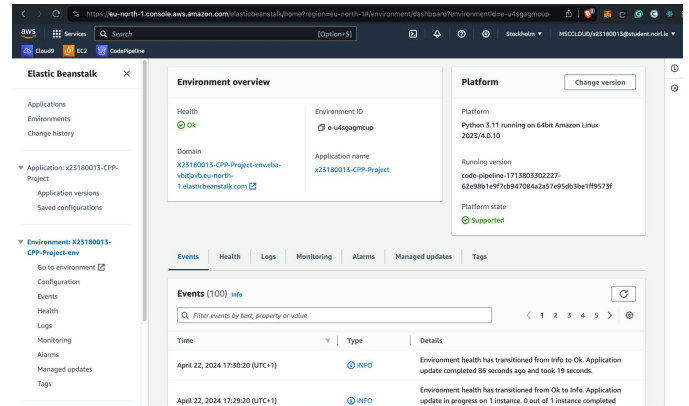


Fig. 3. RoundLab Elastic Beastalk environment

#### B. CodePipeline

CodePipeline is used to automate the process of deploying code from the development stage to the moment it is deployed. Automating the release of our applications is made possible with the help of the Amazon code pipeline. The speed of our software delivery can be improved with this automation. Creating a pipeline is simple and can be done with just a few

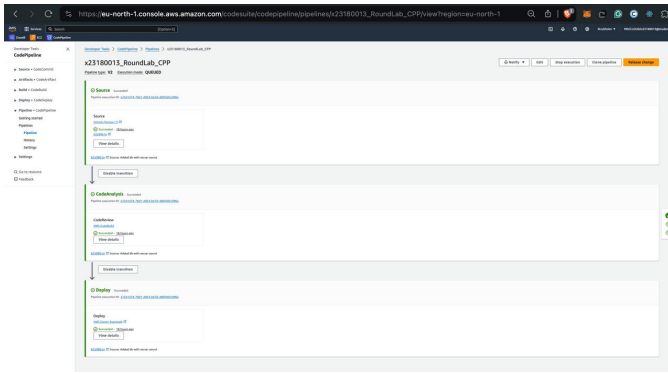


Fig. 4. RoundLab CI/CD pipeline

configurations, unlike the traditional way of using terraform, Jenkins, ansible and some scripting. Proper IAM permissions are needed to create a pipeline. Once that's done, we need to install the AWS CLI and provide the GitHub repository access. Once the setup is done there are three stages 4 in the application's CI/CD pipeline: Source, Static Code Analysis, and Deployment. The Source stage analyzes the source code for bugs, deprecated methods, code smells, and duplication. The Code Analysis stage is where the project build is created and verified against Python coding standards using pylint. The final stage involves using AWS Elastic Beanstalk service for deployment, which automates the deployment process, making it quick and easy.

### C. Simple Storage Service

Amazon Simple Storage Service (S3) [3] is a storage service that provides secure, scalable and reliable storage for crucial data such as websites, archives, backups, user profiles, product images, website photos, and order invoices. Storing high-quality images on a local machine could affect scalability and increase the size of an application package. Therefore, Amazon S3 is a suitable solution for applications looking to store large amounts of data. In addition to storage, Amazon S3 provides features like S3 life cycle, S3 object lock, S3 replication, and S3 batch operations. S3 replication is particularly useful as it minimizes latency by replicating data on different AWS regions. This ensures that data is always available and accessible when needed.

### D. CodeBuild

CodeBuild [4] tool is used for building and testing application code after the development stage. In the cloud environment, it is fully managed, on-demand scalable depending upon usage volume and ready with a set of predefined build environments which can be customized to use our build tool. To satisfy the code analysis stage where pylint command is saved in buildspec file to run the code build against Python coding guidelines, thus helping the application code to be reliable and maintainable. Using pylint helped in the identification of bugs, outdated methods, unused imports and variables, and code quality issues.

### E. CodeDeploy

The application uses Elastic Beanstalk for the deployment process. It takes care of handling the complexities of infrastructure provisioning and management. It uses CodeDeploy under the hood to deploy application versions to the underlying EC2 instances. CodeDeploy takes care of tasks like copying application code to the EC2 instances, executing pre and post-deployment scripts, and managing the deployment lifecycle. Elastic Beanstalk integrates smoothly with CodeDeploy, utilizing its capabilities to ensure dependable and efficient application deployments.

### F. CloudWatch

AWS CloudWatch is a tool that monitors and manages AWS resources and applications. It gives real-time insights into how these resources and applications are performing. It helps users keep track of how much they're using their resources and lets them know when they're deviating from their predefined thresholds. CloudWatch also makes it easy to collect and analyze log data, which is useful for troubleshooting. I have customized my CloudWatch dashboard and metrics as seen in figure 5 to get a better view of AWS resources like Elastic Beanstalk, CodeBuild and S3, setting up all the necessary widgets required for the monitoring or logging of the cloud resources along with setting alerts for any breakdown or billing updates. CloudWatch Events lets users automate their responses to events in the AWS environment, which can help streamline their workflows.

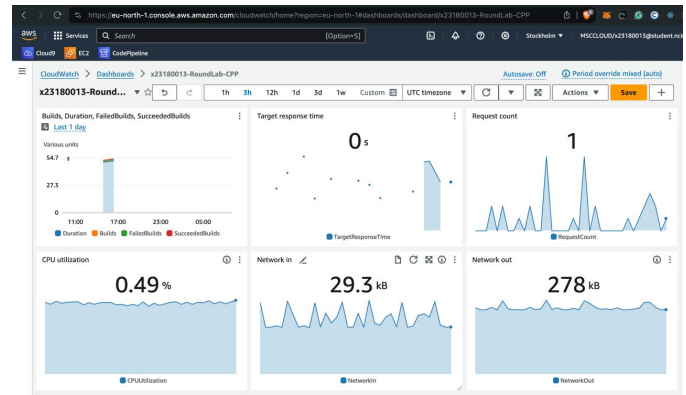


Fig. 5. RoundLab CloudWatch dashboard

### G. Cloud9

Cloud9 is a cloud-based IDE provided by AWS that makes project development easier. It has features such as accessibility, collaboration, and pre-configured environments that allow developers to work from anywhere. With Cloud9, developers can start coding instantly without any setup time. It integrates smoothly with AWS services making deployment and resource management a breeze. The built-in terminal access enables effortless command execution and version control. Customization options enhance productivity by catering to individual preferences. Cloud9 facilitates quick and efficient software



development by streamlining the entire development process and seamlessly integrating with the AWS ecosystem.

#### IV. PYTHON LIBRARY

To achieve code segregation and modular approach I developed and designed a Python library to implement search functionality for my application product collection. It is a lightweight module used to manage product search, get API responses for product lists and render views along with error-handling functions effectively. The library, called search-view-pkg-Anurag09, as seen in figure 7 lets anyone search and retrieve products from the product list.

I started developing the library by extracting the modular code putting it in the library file and creating a directory structure as per the Python library guidelines. Then, create a setup.py script in the project directory to identify key information about the library, such as its name, version, description, and dependencies. The figure shows the library configuration in the code 6 which can be saved in a setup file to build

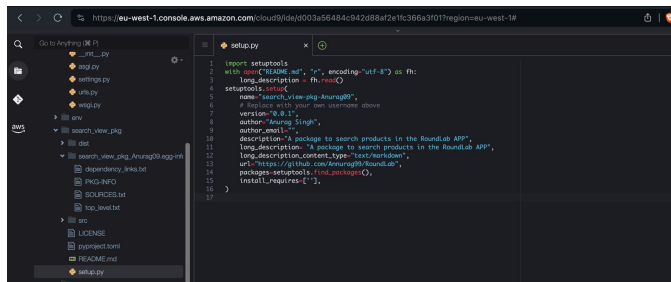


Fig. 6. RoundLab Library configuration

a distribution package with "python setup.py sdist". I then registered to pypi website and created an account, using twine upload dist/\* it got uploaded to pypi site. After uploading, the library is now available for installation via "pip install search-view-pkg-Anurag09". Finally, I tested the library to make sure everything worked correctly before publishing it to the Python community. For the implementation of the library, we need to

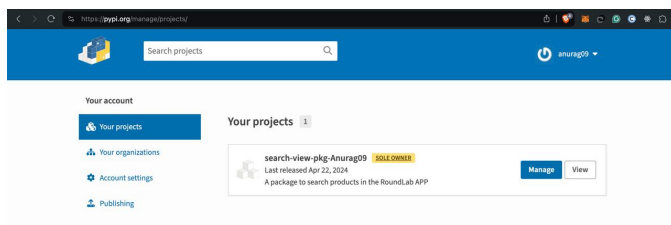


Fig. 7. RoundLab Library published in pypi

import it into our Python file. Then, you can create objects to interact with the product collection. The library also offers different criteria to search for products and retrieve product information. Here's what you can do with search-view-pkg-Anurag09:

- Search for Products: You can use the library to look for products based on their name, category, price range, and more.
- Retrieve Product Information: The library lets you get detailed information about specific products from the product collection.
- Error Handling: The library includes mechanisms to handle any exceptions or errors encountered during product search operations.

The search-view-pkg-Anurag09 makes our application more organized by providing a module dedicated to product-related functions. It makes the process of calling the API and storing the response for rendering the search or retrieving product specifications easy.

#### V. IMPLEMENTATION

The main objective of this report is to implement and study how we can integrate our application with AWS services. We used the cloud9 IDE for coding and debugging. The AWS Learner Academy services were utilized in the application. We used Amazon S3, which is a scalable storage service for storing raw data and static files. I have used S3 bucket to store all pictures including user profiles, products, and website assets, which can be grabbed and rendered in the application by using a URL as shown in figure 8.

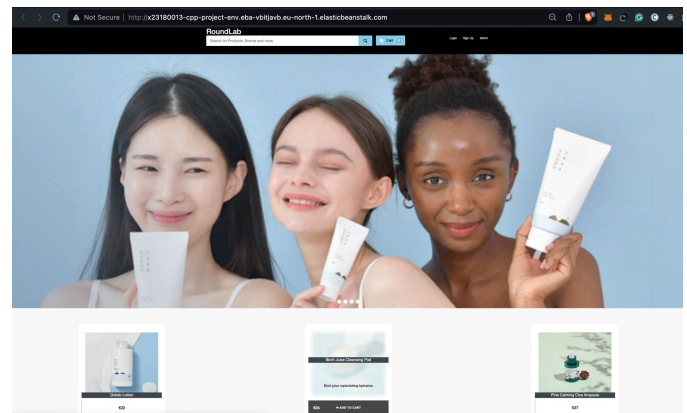


Fig. 8. RoundLab Home page

CodePipeline helped to automate the release process from code commit into GitHub to deployment. Whenever a new commit is pushed, CodePipeline is triggered by GitHub webhooks which detects any source code changes. After the source stage is completed the code analysis stage runs the build of the code against predefined Python linting rules which is defined in pylint. Elastic Beanstalk has made it easy for application deployment by managing infrastructure, load balancer, and Apache servers to serve. It uses AWS CodeDeploy under the hood to deploy application versions to EC2 instances. CodeDeploy handles code copying, executing scripts, and deployment life cycle management. Elastic Beanstalk integrates perfectly with CodeDeploy for quick, dependable and efficient

deployments. I have used CloudWatch to monitor the health of AWS resources like CodeBuild, ElasticBeanstalk and S3.

The application is designed to include functionalities for both admin and customer as per their needs. The admin can log in to the dashboard and manage customers, orders, and products as seen in figure 9. Customers can register, log in,

done have additional roles such as the logistics department in e-commerce. There is also scope for adding an AWS secret manager for storing API secrets to overcome the securing vulnerabilities of storing API keys in environment files. This application will continue to improve and develop by working on user feedback.

## REFERENCES

- [1] D. Webmaster, "The State of EdTech Post-Pandemic - godashlabs.com," <https://godashlabs.com/2021/04/01/state-of-edtech-post-pandemic/>, [Accessed 09-04-2024].
- [2] "What is AWS Elastic Beanstalk? - AWS Elastic Beanstalk — docs.aws.amazon.com," <https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/Welcome.html>, [Accessed 09-04-2024].
- [3] "What is Amazon S3? - Amazon Simple Storage Service — docs.aws.amazon.com," <https://docs.aws.amazon.com/AmazonS3/latest/userguide/Welcome.html>, [Accessed 23-04-2024].
- [4] "What is AWS CodeBuild? - AWS CodeBuild — docs.aws.amazon.com," <https://docs.aws.amazon.com/codebuild/latest/userguide/welcome.html>, [Accessed 23-04-2024].

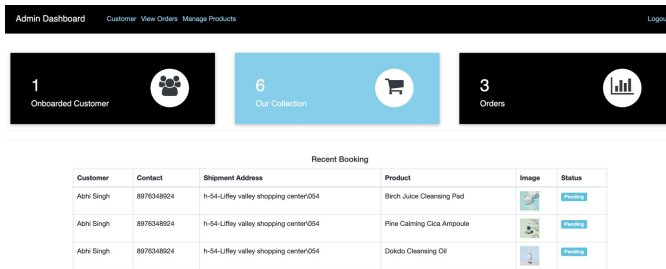


Fig. 9. RoundLab admin dashboard

browse products, add to cart, place orders, track orders 10, and edit their profiles. The live application that has been deployed on the cloud through Amazon Elastic Beanstalk can be accessed by clicking on the link.

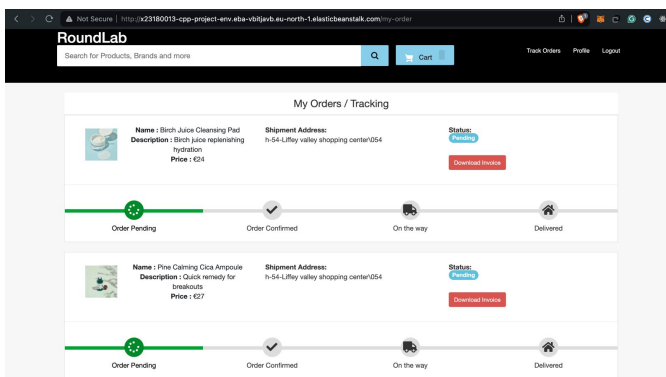


Fig. 10. RoundLab customer order tracking page

## VI. CONCLUSION

In recent years, cloud computing has helped significantly change the software development industry. The various services provided have helped overcome the traditional hardware-dependent limitations by providing on-demand resources and security. This project has helped me understand and get a hands-on experience in the AWS domain. RoundLab is a web application developed and deployed using seven different AWS services - Cloud9, CodeDeploy, CodeCommit, Elastic Beanstalk, S3, CloudWatch and EC2. The application also uses pypi python library to implement the reusable sections of the code. Currently, the application has functionalities designed for only two roles - admin and customer further work can be