FreshBasket

Scalable E-commerce Platform Deployment with Flask on AWS EC2 and RDS

Yuvaraj K	2022506079
Nithish S	2022506099
Anselm Flavian P	2022506101
Kumaran M	2022506125

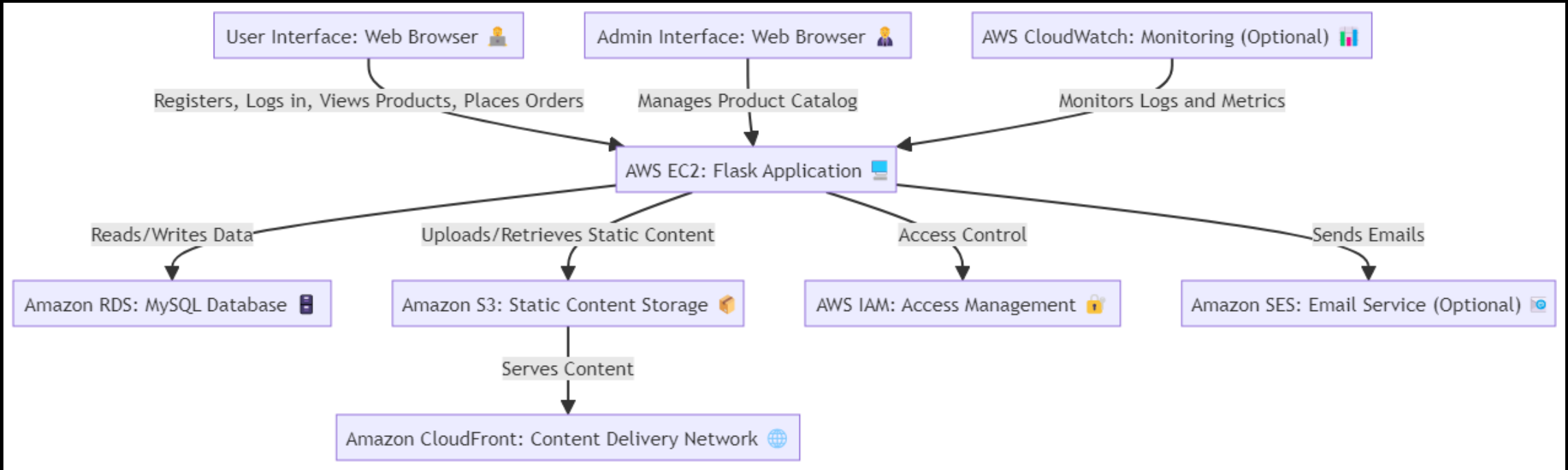
Problem Statement

The "FreshBasket" project is a cloud-native solution that involves

- developing and deploying a scalable e-commerce platform for selling vegetables and fruits.
- using Flask for backend development, AWS EC2 for hosting, and Amazon RDS for database management.
- ensuring high availability, scalability, and efficient management of the platform’s operations.

The project demonstrates how leveraging AWS services can create a robust infrastructure for managing user interactions, product catalogs, and order processing in a seamless manner.

Architecture

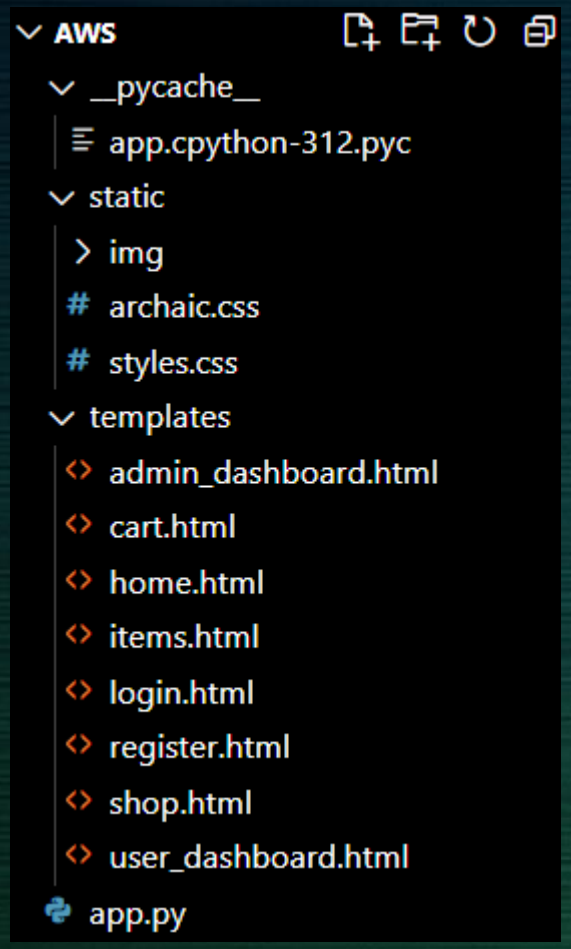


Tools Used

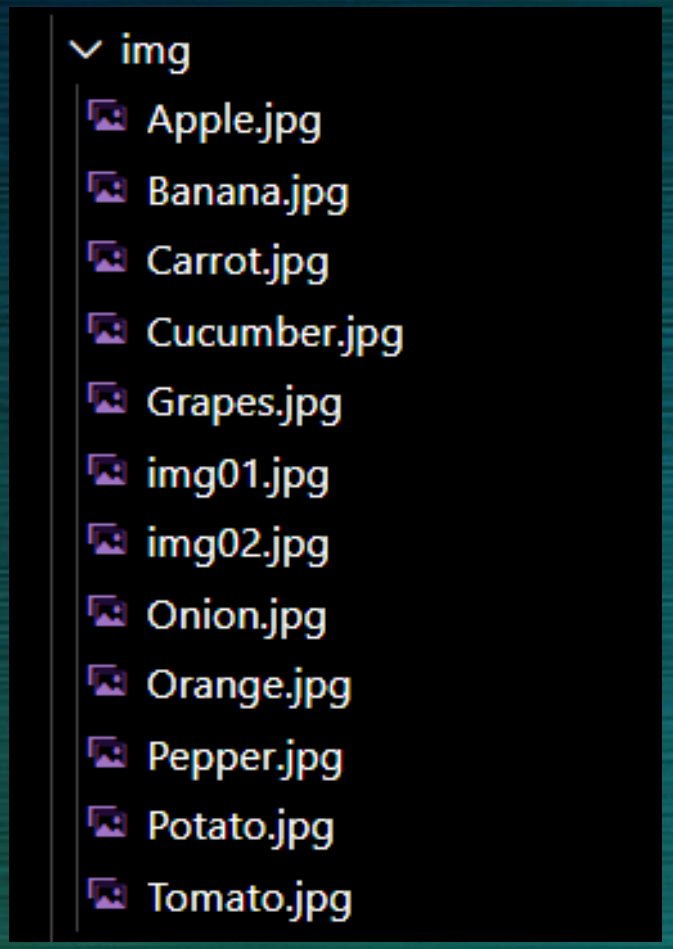
HTML and CSS	Frontend - Structure and Design
Flask	Backend – Data Processing and DB Connection
MySQL	Relational Database Management System
AWS EC2	Hosting the Web Application
AWS RDS	Database Management in AWS
MobaXterm	Secure Shell for EC2 Connection and Hosting

File Structure

Overall Structure – Static and Templates



Images in static/img Folder



Database Tables Structure

users

	Field	Type	Null	Key	Default	Extra
1	id	int(11)	NO	PRI	(NULL)	auto_increment
2	name	varchar(255)	YES	(EMPTY)	(NULL)	(EMPTY)
3	mobile	varchar(20)	YES	(EMPTY)	(NULL)	(EMPTY)
4	email	varchar(255)	YES	(EMPTY)	(NULL)	(EMPTY)
5	password	varchar(255)	YES	(EMPTY)	(NULL)	(EMPTY)
6	address	text	YES	(EMPTY)	(NULL)	(EMPTY)

items

	Field	Type	Null	Key	Default	Extra
1	item_id	int(11)	NO	PRI	(NULL)	auto_increment
2	item_name	varchar(255)	NO	(EMPTY)	(NULL)	(EMPTY)
3	price	decimal(10,2)	NO	(EMPTY)	(NULL)	(EMPTY)

Database Tables Structure

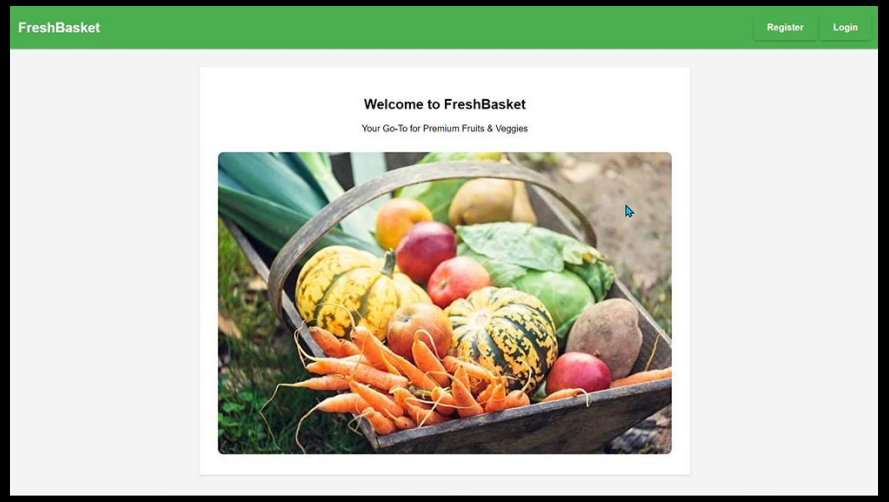
orders

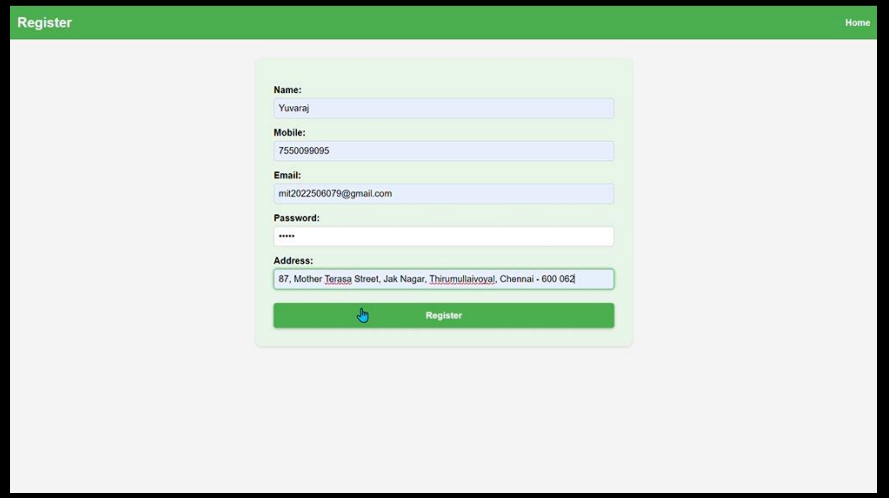
	Field	Type	Null	Key	Default	Extra
1	id	int(11)	NO	PRI	(NULL)	auto_increment
2	user_id	int(11)	YES	MUL	(NULL)	(EMPTY)
3	delivery_address	text	YES	(EMPTY)	(NULL)	(EMPTY)
4	payment_method	varchar(50)	YES	(EMPTY)	(NULL)	(EMPTY)
5	total_price	decimal(10,2)	YES	(EMPTY)	(NULL)	(EMPTY)
6	status	varchar(50)	YES	(EMPTY)	(NULL)	(EMPTY)
7	order_date	datetime	YES	(EMPTY)	(NULL)	(EMPTY)

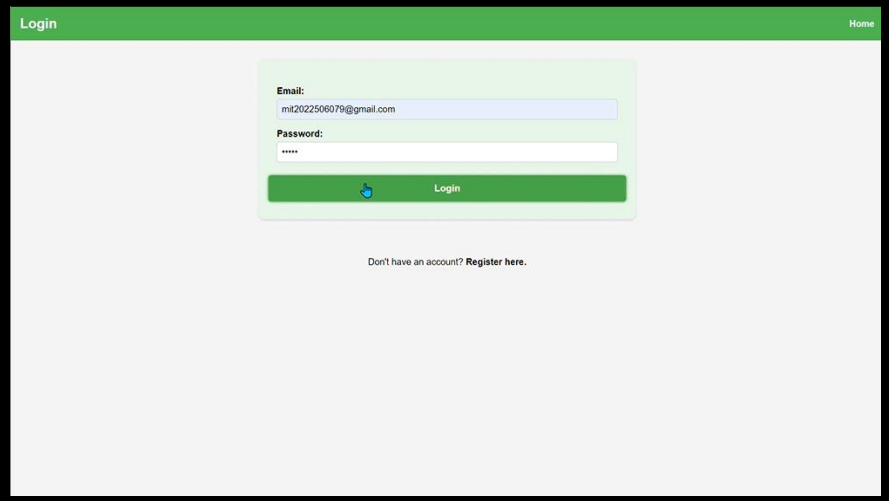
order_items

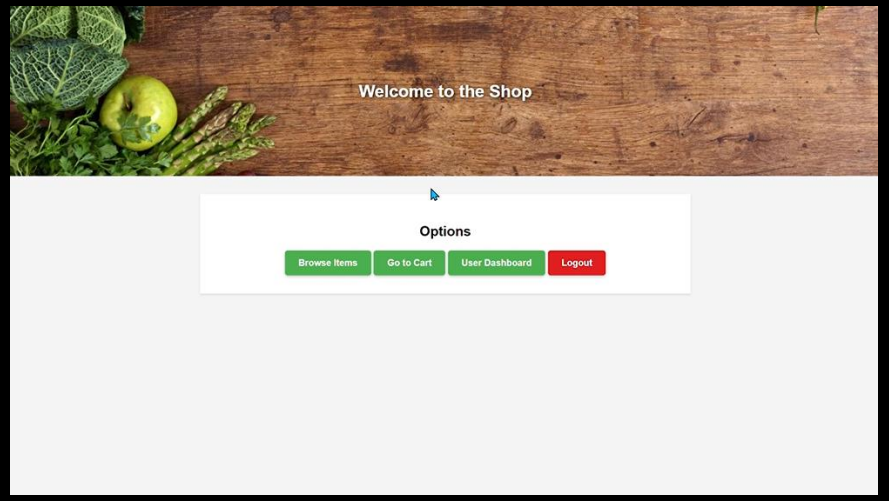
	Field	Type	Null	Key	Default	Extra
1	id	int(11)	NO	PRI	(NULL)	auto_increment
2	order_id	int(11)	YES	MUL	(NULL)	(EMPTY)
3	item_name	varchar(255)	YES	(EMPTY)	(NULL)	(EMPTY)
4	item_price	decimal(10,2)	YES	(EMPTY)	(NULL)	(EMPTY)
5	item_quantity	int(11)	YES	(EMPTY)	(NULL)	(EMPTY)

Output Screenshots

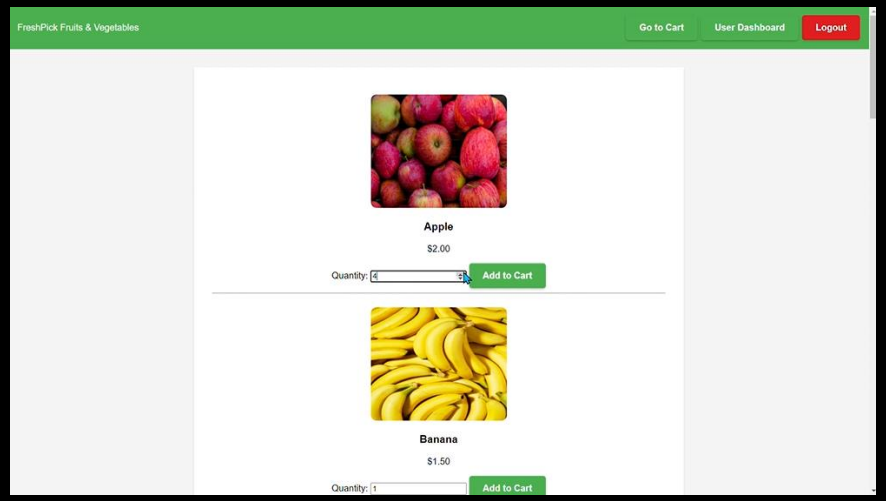


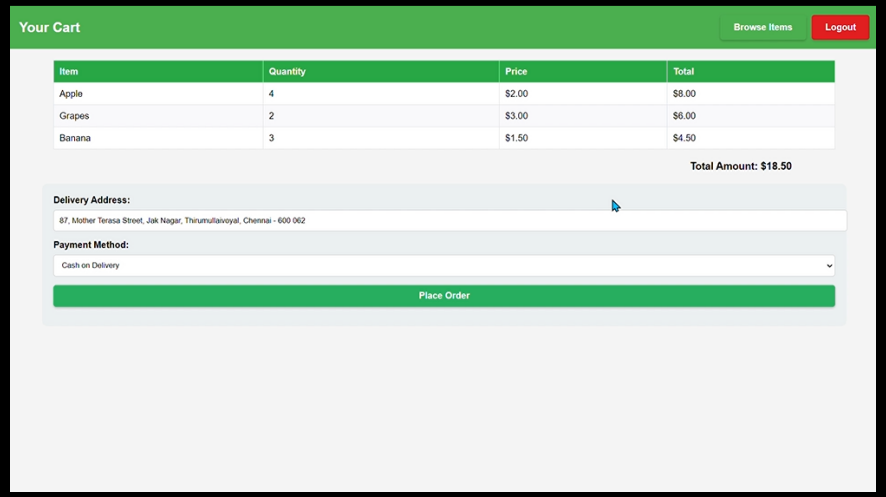


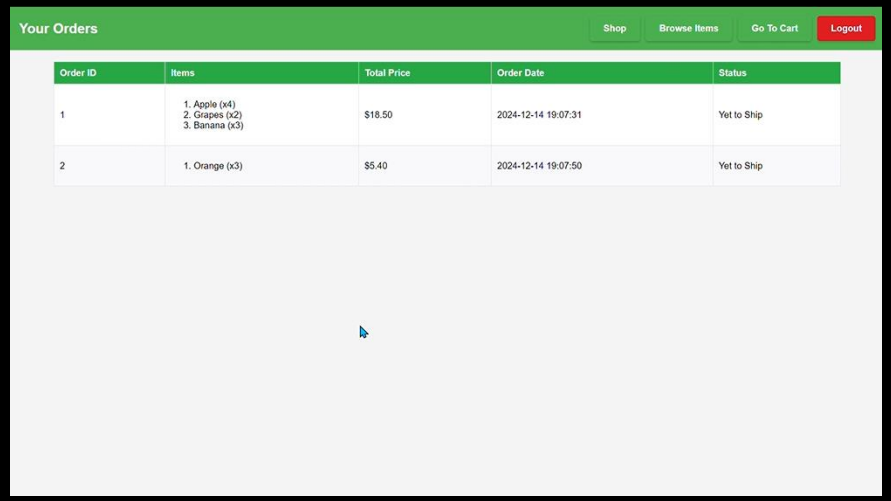


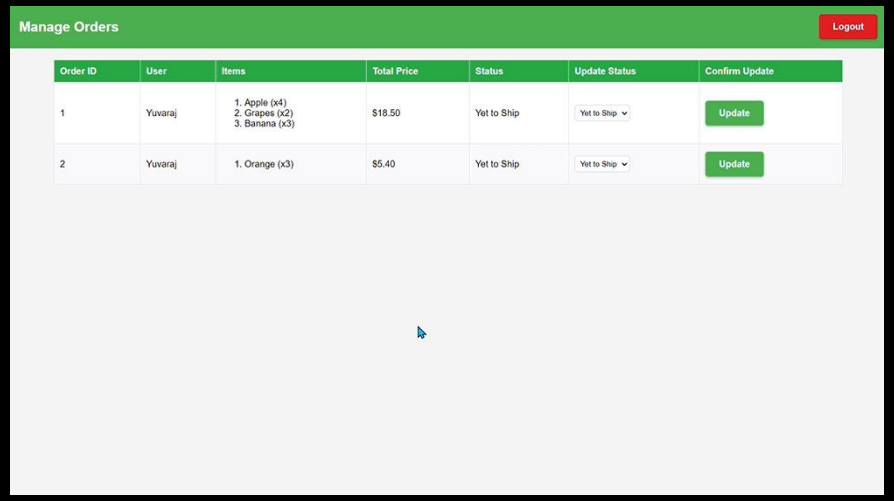


Output Screenshots









Current Limitations

The FreshBasket application for grocery shopping is reliable, fast, and scalable thanks to the integration with AWS. However, it does have limitations that can be addressed.

- Monitoring the performance of the application with increase in traffic – can be done with Amazon CloudWatch. Not implemented currently due to the smaller scale of the application.
- An email service to verify emails with OTPs – can be implemented with Amazon Simple Email Service. This could also be accomplished in Flask itself.
- Design improvements can be made to provide a more appealing look that still maintains the simplicity and ease of use of the application.

Conclusion

The FreshBasket project showcases the deployment of a scalable and efficient e-commerce platform using AWS.

- Integrating Flask with MySQL RDS, ensures smooth data management and secure backend operations.

- The EC2 instance, coupled with MobaXterm, enables seamless remote management and deployment of the application.

From database setup to frontend development and deployment, each milestone was carefully executed to ensure reliability and performance. Overall, the project demonstrates the ability of AWS services to support dynamic, scalable web applications in real-world scenarios.