Project Proposal

Group 5
Archer Zhou
Junqiao Qu
Kessel Zhang

We intend to develop an online shopping website with the support of AWS service. The website provides customers with basic functions for shopping and receiving orders function for sellers.

Potential User Story/Task

User Story 1:

The user could create an account with sufficient personal information.

User Story 2:

The user could log in using an existing account

User Story 3:

User add items to shopping cart (saved in account)

User Story 4:

Users could check out the cart, pay for orders (PayPal API).

User Story 5:

Items could be posted on the website

User Story 6:

Orders could be saved in a database, with transaction details

User Story 7:

The user could review the history order.

User Story 8:

The user could do an online chat on this website.

AWS Service

AWS SNS

We choose AWS SNS because it provides message delivery from publishers to subscribers.

We could use this service to push notifications for order confirmation/status to customer/seller email addresses.

AWS Secrets Manager

AWS Secrets Manager helps us protect the keys we need to access applications, services.

We could use this service to save API keys, and important user information (username/password).

AWS API Gateway

API Gateway has powerful, flexible authentication mechanisms, such as AWS Identity and Access Management policies, Lambda authorizer functions, and Amazon Cognito user pools.

We could use this service to secure networks requests.

AWS S3

We choose S3 because it is a simple storage service and this is a small project which means it will fit us well.

We could use this service to store images of items, static resources.

(AWS DynamoDB)

We choose DynamoDB because it provides NoSQL which reduces the complexity. It also has built-in security, backup, and restore features

We could use this as our online database, for order information storage

AWS Elastic Beanstalk

We chose this service because it can automatically handle deployments including capacity prepositioning, load balancing, auto-scaling, and application performance monitoring.

We will use this for back-end code.

AWS Lambda

We choose AWS Lambda because it could let the code run without provisioning or managing servers and can build data-processing triggers for AWS S3 and DynamoDB we mentioned above.

Use this for some service to build a serverless backend.

Delivery Model

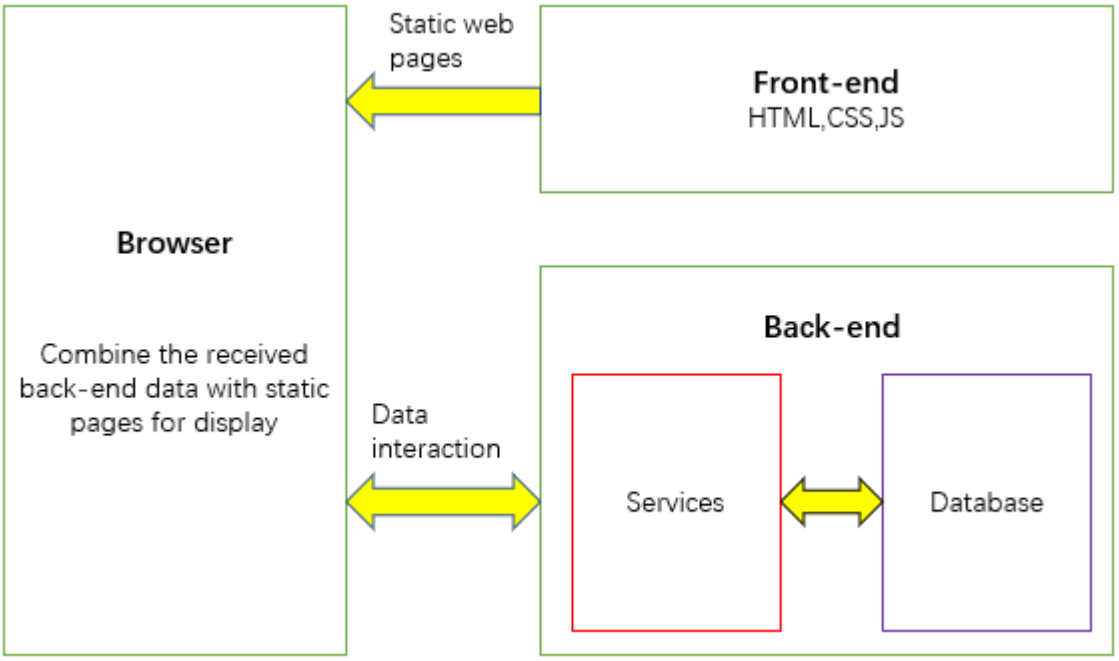
The website contains two parts. The front-end provides a user interaction interface and may use the framework REACT. It will use AWS Elastic Beanstalk and it will be a PaaS model. In this model, we have control of our application and data. And the cloud service provider will take the responsibility of managing runtime, middleware, OS, networks and IT resources. It provides a ready-to-use environment for our application.

And the back-end service will use AWS Lambda to build a service to manage order information and other data. It will be a FaaS Model. In this model, we have the control of function code and do not need to manage applications, platforms for running and other IT resources.

Deployment Model

Since our website, application, service mainly uses AWS computing resources. So we use the Public Clouds deployment model. There are. First, we do not have physical IT resources. We need a publicly accessible cloud environment to support our website. So public clouds are our best choice for this project.

Interact Diagram



Plan

Milestone	Tentative deadline	Completeness	Delievarables
reading document	2/20	5%	N/A
user interface design (Figma)	2/23	20%	blueprint of webpages
architecture design	2/27	35%	design document
front-end back-end development	3/25	85%	web project
integration test	3/27	100%	test report
project document and report	4/4	N/A	final presentation and report

REF:

https://docs.aws.amazon.com/sns/latest/dg/welcome.html

https://docs.aws.amazon.com/secretsmanager/latest/userguide/intro.html

https://docs.aws.amazon.com/apigateway/latest/developerguide/welcome.html

https://docs.aws.amazon.com/AmazonS3/latest/userguide/Welcome.html

https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Introduction.html

https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/Welcome.html

https://docs.aws.amazon.com/lambda/latest/dg/welcome.html