Project Proposal

Member

Archer Zhou
Junqiao Qu
Kessel Zhang

Overview

We plan to develop an online shopping website with the support of cloud computing service. This website will be a Business-To-Customer. It will provide basic functions for shopping and receiving orders. Customers could look through items on the website, make a payment and receive order confirmation after payment. The details of order will be saved into the database.

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:

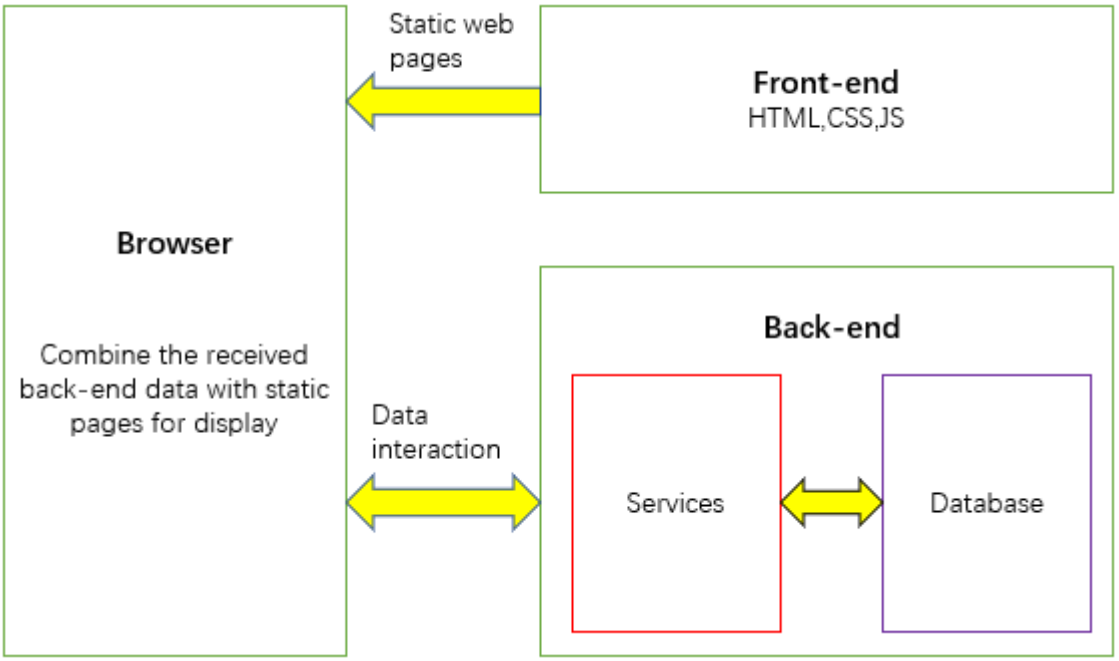
The user could review the history order.

(User Story 8):

The user could do an online chat on this website.

Description

Interact Diagram



Cloud Service

AWS SNS
AWS Secrets Manager
AWS API Gateway
AWS DynamoDB
AWS Elastic Beanstalk
AWS Lambda

Mechanisms

General

AWS SNS

We choose AWS SNS because we need a simple way to notify users by email, text. We could use this service to push notifications for order confirmation/status to customer/seller email addresses.

AWS Secrets Manager

We need a safe way to manage our users' account information, and API keys of some service.
AWS Secrets Manager helps us protect the keys we need to access applications, services and important user information (username/password).

Network

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 network requests.

Storage

AWS S3

We choose S3 because we need a simple storage service for image of items. And this 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

Compute

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.

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	100%	final presentation and report

Delivery and Deployment

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. Second, our project is designed for general commercial use. So the public clouds will provide sufficient safety for data. We trust them on the service level of agreement. In all, public clouds are our best choice for this project.

The benefit is that the cloud provider is fully responsible for physical IT resources, so we don't need to worry about the configuration. We just need to set up our application code and use the API of it. It is free of use and powerful enough for our project.

The drawback is multi-tenancy and data security. We do not have the control of data in our own device. We need to trust the cloud provider for data security.

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.

Reference

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