A REPORT

ON

BACK END DEVELOPMENT

BY

Somya Upadhyay 2017B5A40962P

AT

MySmartPrice, Bangalore

A Practice School-II station of

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI

(December, 2021)

A REPORT

ON

BACK END DEVELOPMENT

BY

Name	ID.No.	Discipline
Somya Upadhyay	2017B5A40962P	M.Sc. Physics and B.E. Mechanical

Prepared in fulfilment of the

Practice School-II Course

AT

MySmartPrice, Bangalore

A Practice School-II station of

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI

(December 2021)

Acknowledgements

I am using this opportunity to express my gratitude to everyone who supported me throughout the project. I am thankful for their aspiring guidance, invaluably constructive criticism and friendly advice during the project work. I am sincerely grateful to them for sharing their truthful and illuminating views on several issues related to the project.

I express my warm regards to, **Mr Sitakanta Ray (***Head and Founder)* for his mentorship, support and encouragement.

I am gratetful to **Mr Arun Chinnanchamy** (*Professional Incharge of the project and Head)*for his support and guidance at MySmartPrice.

I would also like to thank the HR and IT departments for providing me all the facilities and softwarenecessary for the accomplishment of this project.

I would also like to thank my PS Faculty **, Mrs. Monali Mavani** for his support in completion of this project.

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE

PILANI (RAJASTHAN)

Practice School Division

Station: MySmart Price **Centre .**Bangalore

Duration 5 months **Date of Start** 19th July,2021

Date of Submission 10th December ,2021

Title of the Project: Back end Development and cloud deployment

ID No./ 2017B5A40962P

Name(s)/ Somya Upadhyay

Discipline(s)/of M.Sc. Physics and BE Mechanical Engineering

the student(s) BE Mechanical Engineering

Name(s) and Mr Arun Chinnachamy

designation(s) Head of Engineering

of the expert(s): Bright Champs

Name(s) of the

PS Faculty: Prof. Monali Mavani

Key Words: Cloud Computing Services, SaaS, Database management, REST API, Messaging,

Project Areas: IT Development

Abstract:

The project aims to create a new SAAS product. The production is streamlined using tools of organising software development such as popular models like Agile, open-source APIs, frameworks and decoupled micro-services and users by cloud computing and programming API. The product is meant for mid-scale e-commerce companies online sales support, as well as to enable other features on e-commerce websites.

The microservices made by me during my development were first made in Python Flask and later converted to Express (node.js). The RESTFUL API architecture was followed. The Database was centralised on the AWS cloud based service RDS.

Signature(s) of Student(s) **Signature of PS Faculty**

Date 14/12/2021 **Date 14/12/2021**

Table of Contents

COVER……………………………………………………………..1

TITLE PAGE…………………………………………………….2

ACKNOWLEDGEMENTS……………………………………..3

ABSTRACT SHEET…………………………………………..4-5

INTRODUCTION……………………………………………8-9

MAIN TEXT………………………………………………..10-14

Basics of Cloud Computing Services

Stages of Production

POST MID-SEMESTER…………………………………..……..15-25

Deployment - using Web server Nginx and WSGI server Gunicorn

Sub-features integration -SQS and webhooks

Google APIs

Learning Outcome: SDLC

Comparing micro frameworks: Flask vs Express on node.js

Examples

SUMMARY & CONCLUSION …………………........................26-27

APPENDIX……………………………………………...….28-31

REFERENCES……………………………………………..…..32

INTRODUCTION

My Smart Price

MySmartPrice is an online engine that compares the prices of mobiles, books, electronics, cameras, and appliances, thereby helping consumers make the best decisions. **Sulakshan Kumar and Sitakanta Ray are the founders of MySmartPrice.**

MySmartPrice is a high-growth venture-backed start-up that runs the largest comparison shopping site in India, with over **20 Million user visits per month**. They help people quickly find 'what to buy' and 'where to buy from' and help them make better and wiser purchases.

Currently, MySmartPrice is working to **extend SaaS products** to medium scale e-commerce businesses.

Retainiq

Retainiq is a new product in affiliation with MSP for an internet-based service for clients. Clients are people who hire third-party tech solutions for their e-commerce sales. We offer interactive and data intelligence services to help small and medium-sized companies compete with e-commerce giants such as Amazon and Flipkart on an equal footing.

In the process, I have been exposed to the functioning of a company working on a fixed SDLC model.

My Role

My assignments involve creating REST APIs to incorporate features like

- Automate communication with end-point customers on SMS (outbound and inbound)
- Integrate AWS features such as SQS and Kinesis to ensure the smooth running of SaaS with enhanced performance through data analysis in the future
- DataBase management
- Using Google APIs for navigating huge amounts of data.
- Using Javascript to create an interactive interface

MAIN TEXT

1. Basics of Cloud Computing Services

Overview

Before the availability of Cloud Computing services, the process of web app hosting was On-Premise.

On-premise production refers to independent development where all functioning is managed in house from Data to networking to OS.

Infrastructure as a Service: IaaS refers to the sector that enables the fundamental functioning of storage, network, and computational resources via the internet.

Platform as a Service: PaaS is a deployment service model that delivers software and hardware tools needed to run and develop applications.

Software as a Service: Software as a Service is different in that it speaks to the accessing of software via the internet.

SaaS or Application Cloud is built on Service cloud or Web Service components (eg: Google maps, Microsoft Azure, etc). These are built on Infrastructure Cloud (Network, Computing and Middleware handler).

1.1 SaaS

Software as a service is a way of delivering centrally hosted applications over the internet as a service.

I work on providing micro-services and database handling for a SaaS.

This integration provides features for e-commerce businesses with a personalised interactive experience for its Client customers.

1.2 About IaaS tools

Amazon Web Services, Inc. (**AWS**) is a subsidiary of Amazon providing on-demand cloud computing platforms and APIs to users. RetainIQ uses IaaS cloud services like:

EC2: Elastic cloud computing: secure web scaling.

RDS: Relational Database server: database management service.

KDS: Kinesis Data Stream: real-time data streaming service from websites and events.

SQS: Simple Queue Service: To send/store/ receive messages between software components online.

1.3 About CPaaS tools

Twilio is an American cloud communications platform as a service (CPaaS) company that allows software developers to programmatically make and receive phone calls, send and receive text messages, and perform other communication functions using its **web service APIs**.

Twilio provides a bulk messaging service that I use for my API and its inbound(incoming message) **webhook** that could help clients set up *interactive communication system/ automated help-desk chatbot* with no need to code or handle scaling in house.

2. Microframework for REST API

APIs and Frameworks

An a**pplication programming interface (API**) is a connection between computers or between computer programs. It is a type of software interface, offering a service to other pieces of the software. An API is a mechanism that enables an application or service to access a resource within another application or service.

A REST API is an API that conforms to the design principles of the REST or *representational state transfer* architectural style.

I used **flask**, a python based microframework to make a scalable API.

Flask offers a range of inbuilt packages to help developers create easily deployable Web apps and REST APIs.

3. Stages of Production

3.1 Relational Database Diagram

First, a relational Database diagram frame is created and populated with sample data to try the SaaS micro-services. For this purpose, My teammate and I made a database diagram using MySQL workbench on an AWS RDS connection.

3.2 Prototype Algorithm and code

The SaaS performs a vast list of tasks integrated by differentiated APIS that can communicate while decoupled through HTTP requests. These APIs are further broken down into functions that share data to perform a task in coordination. My API on being requested performs two functions:

1. Collect template data from RDB
2. Call the provider and update the log

3.3 Testing

The SMS API provides 2 functionalities to test

1. Sms_one(sender,receiver,template,data) :

Sends out 1 message per query and updates log.

2. show_log( ) :

Returns the entire log of all events with time stamp.

For testing, we use Postman, a developer and tester friendly web browser with organised requests and example tables storing the raw data returned by requests.

3.4 Uploading Project to Cloud server

Before deploying, I must access the AWS server through the command line to upload my code.

Secure Shell (SSH) is a cryptographic network protocol for operating network services securely over an unsecured network. Typical applications include remote command-line, login, and remote command execution, but any network service can be secured with SSH.

SSH can be done through clients like **PuTTY**.

To transfer files we could either upload code to **GitHub** and clone it to the server or transfer it directly through any SSH client like **WinSCP**.

For my server, I used the command prompts and PuTTY. I uploaded the project using WinSCP because of its easy drag and drop feature.

Post-Mid-semester

3.5 Deployment

Three common building blocks when deploying a Python web application to production are:

1. 8A webserver (like Nginx)
2. A WSGI application server (like Gunicorn)
3. Application using a framework (like Flask)

We will use **Gunicorn** as a WSGI server to communicate with our **flask app** and **Nginx** as a proxy server between the Gunicorn server and the client.

We need Gunicorn between flask and Nginx because the flask development server although good for debugging is weak and will not stand in production, so we need Gunicorn as a WSGI server to communicate with the flask.

4. Feature Integration

4.1 Real-time Data flow

To integrate a central relational database with other systems by streaming its modifications through Amazon Kinesis.

One important detail of this method is that the consumers won't receive SQL queries. Those can be exposed too, but in general, observers won't be very interested in SQL unless they maintain a SQL-compatible replica of the data themselves. Instead, they will receive modified entities (rows) one by one. The benefits of this approach are that consumers do not need to understand SQL and the single source of truth does not need to know who will be consuming its changes.

4.2 Webhook listening for inbound messages

To listen for messages sent by users of the eCommerce platforms(enquiries/complaints), we will set up webhooks that trigger an API to take the desired action(respond, update log, forward to customer care, etc).

4.3 Decoupling and Scaling: SQS

Amazon Simple Queue Service (SQS) is a fully managed message queuing service that enables you to decouple and scale microservices, distributed systems, and serverless applications. SQS eliminates the complexity and overhead associated with managing and operating message-oriented middleware and empowers developers to focus on differentiating work.

Using SQS, you can send, store, and receive messages between software components at any volume, without losing messages or requiring other services to be available.

5. SDLC

5.1.1 Planning

This is the first phase in the systems development process. It identifies whether or not there is the need for a new system to achieve a business"s strategic objectives. This is a preliminary plan (or a feasibility study) for a company"s business initiative to acquire the resources to build on an infrastructure to modify or improve a service.

The company might be trying to meet or exceed expectations for their employees, customers, and stakeholders. This step aims to find out the scope of the problem and determine solutions. Resources, costs, time, benefits and other items should be considered at this stage.

5.1.2 Systems Analysis and Requirements

The second phase is where businesses will work on the source of their problem or the need for a change. In a complex situation, possible solutions are submitted and analysed to identify the best fit for the project's ultimate goal(s). Now, the teams consider the functional requirements of the project or solution. It is also where system analysis takes place—or analysing the needs of the end-users to ensure the new system can meet their expectations. Systems analysis is vital in determining what a business"s needs are, how they can be met, who will be responsible for individual pieces of the project, and what sort of timeline should be expected.

There are several tools businesses can use specific to the second phase. They include:

- CASE (Computer Aided Systems/Software Engineering)
- Requirements gathering

- Structured analysis

5.1.3 Systems Design

The third phase describes, in detail, the necessary specifications, features and operations that will satisfy the functional requirements of the proposed system, which will be in place. This is the step for end-users to discuss and determine their specific business information needs for the proposed system. It"s during this phase that they will consider the essential components (hardware and software), structure (networking capabilities), processing and procedures for the system to accomplish its objectives.

5.1.4 Development

The fourth phase is when the real work begins—particularly when a programmer, network engineer, and database developer are brought on to do the significant work on the project. This work includes using a flow chart to ensure that the process of the system is organised correctly. The development phase marks the end of the initial section of the process. Additionally, this phase signifies the start of production. The development stage is also characterised by installation and change. Focusing on training can be a huge benefit during this phase.

5.1.5 Integration and Testing

The fifth phase involves systems integration and system testing (of programs and procedures)—customarily carried out by a Quality Assurance (QA) professional—to determine if the proposed design meets the initial set of business goals. We may repeat testing, specifically to check for errors, bugs and interoperability. This testing will be performed until the end-user finds it acceptable. Another part of this phase is verification and validation, both of which will help ensure the program"s successful completion.

5.1.6 Implementation

The sixth phase is when the majority of the code for the program is written. Additionally, this phase involves the actual installation of the newly-developed system. This step puts the project into production by moving the data and components from the old system and placing them in the new system via a direct cutover. While this can be a risky (and complicated) move, the cutover typically happens during off-peak hours, thus minimising the risk. Both system analysts and end-users should now see the realisation of the project that has implemented changes.

5.1.7 Operations and Maintenance

The seventh and final phase involves maintenance and regular required updates. This step is when end users can fine-tune the system, if they wish, to boost performance, add new capabilities or meet additional user requirements.

5.2 Waterfall Model

Implementing this project management methodology requires a lot of up-front planning and preparation. A big part of waterfall project management is creating an airtight project plan so your team clearly understands the project requirements—and restraints—before they get started on the work. That’s because there isn't a lot of room for variation, adaptability, or error once a waterfall project is set in motion.

With careful planning, you can successfully achieve your end product with clear, predictable workflows. This project methodology is great for time management and progress tracking, though it is less flexible than other models.

5.3 Agile Model

Following are the phases in the Agile model are as follows:

1. Requirements gathering: In this phase, you must define the requirements. You should explain business opportunities and plan the time and effort needed to build the project. Based on this information, you can evaluate technical and economic feasibility.

2. Design the requirements: When you have identified the project, work with stakeholders to define requirements. You can use the user flow diagram or the high-level UML diagram to show the work of new features and show how it will apply to your existing system.

3. Construction/ iteration: When the team defines the requirements, the work begins. Designers and developers start working on their project, which aims to deploy a working product. The product will undergo various stages of improvement, so it includes simple, minimal functionality.

4. Testing: In this phase, the Quality Assurance team examines the product's performance and looks for the bug.

5. Deployment: In this phase, the team issues a product for the user's work environment.

6. Feedback: After releasing the product, the last step is feedback. In this, the team receives feedback about the product and works through the feedback.

Agile Testing Methods:

- Scrum is an agile development process focused primarily on managing tasks in team-based development conditions.
  - Scrum Master: The scrum can set up the master team, arrange the meeting and remove obstacles to the process
  - Product owner: The product owner makes the product backlog, prioritises the delay, and is responsible for distributing functionality on each repetition.
  - Scrum Team: The team manages its work and organises it to complete the sprint or cycle.
- Crystal- has three concepts.
  - Chartering: Multi activities are involved in this phase, such as making a development team, performing feasibility analysis, developing plans, etc.
  - Cyclic delivery: under this, two more cycles consist, these are:
    - The team updates the release plan.
    - The integrated product delivers to the users.
  - Wrap up: This phase performs deployment post-deployment according to the user environment.
- Dynamic Software Development Method(DSDM)
- Feature Driven Development(FDD)
- Lean Software Development
- eXtreme Programming(XP)

6. Comparing EXPRESS with Flask

6.1 What is node.js?

Node.js is an open-source, cross-platform runtime environment that allows developers to create all kinds of server-side tools and applications in JavaScript. The runtime is intended for use outside of a browser context (i.e. running directly on a computer or server OS). The environment omits browser-specific JavaScript APIs and adds support for more traditional OS APIs including HTTP and file system libraries.

6.2 What is Express?

Express the most popular *Node* web framework and is the underlying library for severalother popular Node web frameworks. It provides mechanisms to:

- Write handlers for requests with different HTTP verbs at different URL paths (routes).
- Integrate with "view" rendering engines to generate responses by inserting data into templates.
- Set standard web application settings like the port to use for connecting, and the location of templates that are used for rendering the response.
- Add additional request processing "middleware" at any point within the request handling pipeline.

While *Express* itself is relatively minimalist, developers have created compatible middleware packages to address almost any web development problem. There are libraries to work with cookies, sessions, user logins, URL parameters, POST data, security headers, and *many* more.

You can find a list of middleware packages maintained by the Express team at Express Middleware (along with a list of some popular 3rd party packages).

❖ **Flask**	❖ **Express JS**
❖ is a Python-based Micro Framework.	❖ is javascript based Microframework.
❖ Slower for more requests	❖ Express is much faster.
❖ Different syntax than front-end. It needs specialisation separately for the frontend and backend.	❖ Startups with fewer engineers can benefit from using the same language in frontend and backend.
❖ More readable syntax,	❖ A not-beginner friendly language
❖ fewer development libraries.	❖ vast ecosystem of libraries

7. GOOGLE APIs

Google APIs are application programming interfaces (APIs) developed by Google, allowing communication with Google Services and their integration to other services.

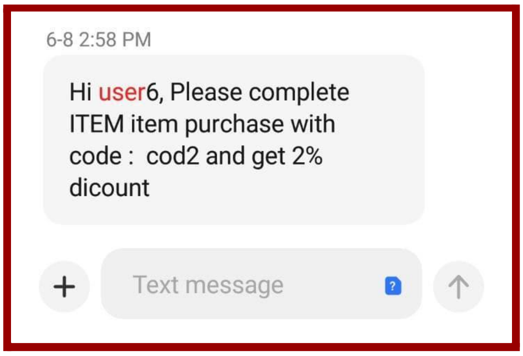
There are client libraries in various languages which allow developers to use Google APIs from within their code, including Java, JavaScript, Ruby, .NET, Objective-C, PHP and Python.

We used a google API that searches through Linkedin results to consolidate relevant data for our clients and save it to the central database.

The **Custom Search JSON API** lets you develop websites and applications to retrieve and display search results from Programmable Search Engine programmatically. With this API, you can use RESTful requests to get either **web search** or **image search** results in JSON format.

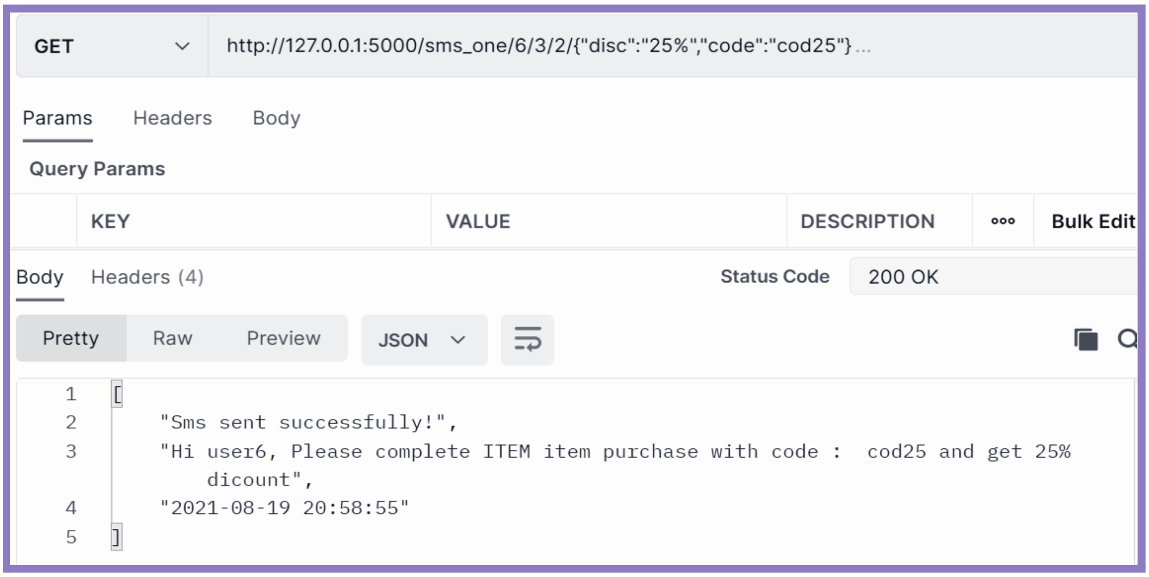
8. Example requests and test data

A) SMS_one/<CustomerId>/<templateId>/<ShopId>/(Payload):


6-8 2:58 PM

Hi user6, Please complete ITEM item purchase with code : cod2 and get 2% dicount

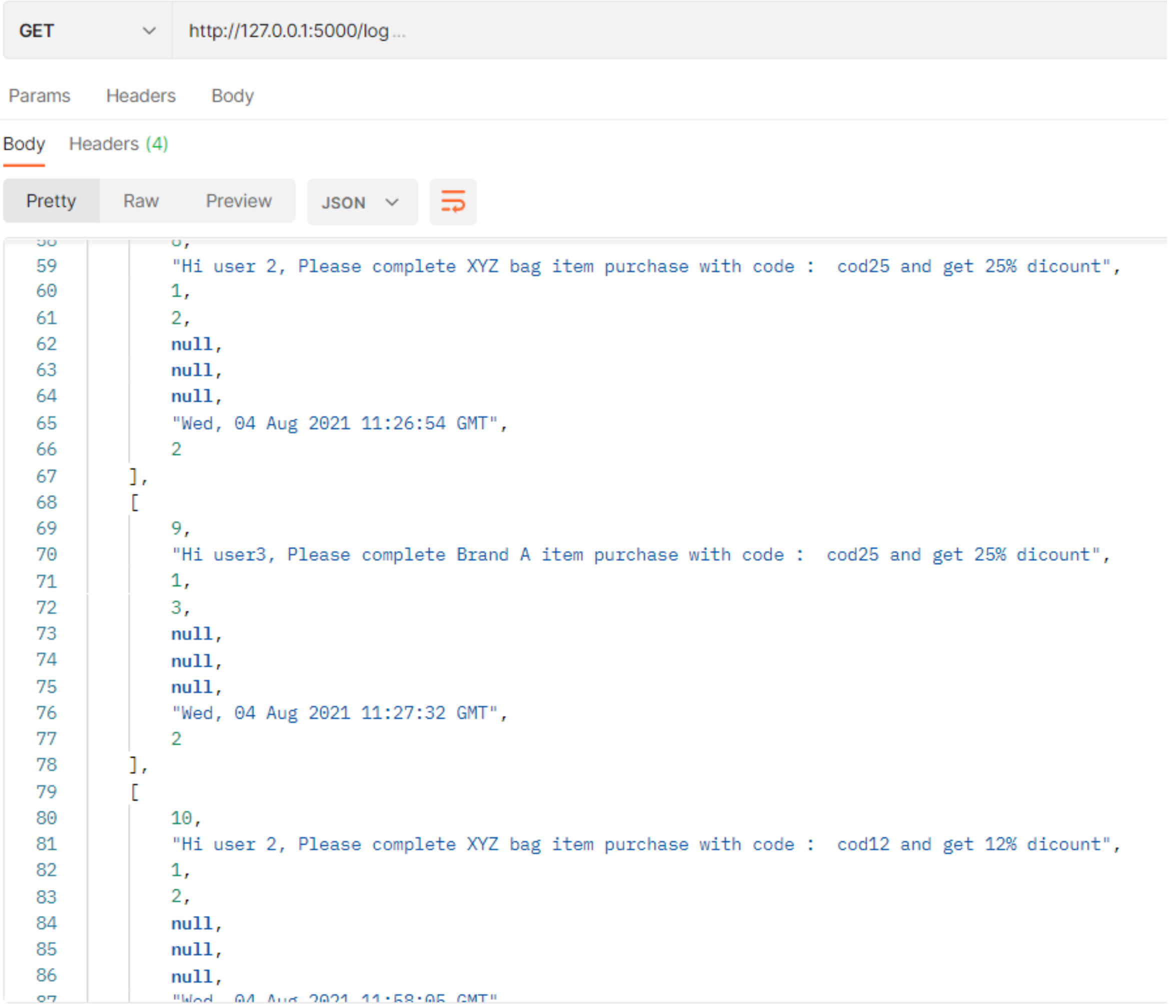
Mobile SMS snap


GET http://127.0.0.1:5000/sms_one/6/3/2/{"disc":"25%","code":"cod25"}...

Status Code 200 OK

```
[
    "Sms sent successfully!",
    "Hi user6, Please complete ITEM item purchase with code :  cod25 and get 25%
        dicount",
    "2021-08-19 20:58:55"
]
```

Request example for sms_one function



GET http://127.0.0.1:5000/log ...

```
        8,
        "Hi user 2, Please complete XYZ bag item purchase with code :  cod25 and get 25% dicount",
        1,
        2,
        null,
        null,
        null,
        "Wed, 04 Aug 2021 11:26:54 GMT",
        2
    ],
    [
        9,
        "Hi user3, Please complete Brand A item purchase with code :  cod25 and get 25% dicount",
        1,
        3,
        null,
        null,
        null,
        "Wed, 04 Aug 2021 11:27:32 GMT",
        2
    ],
    [
        10,
        "Hi user 2, Please complete XYZ bag item purchase with code :  cod12 and get 12% dicount",
        1,
        2,
        null,
        null,
        null,
        "Wed, 04 Aug 2021 11:58:05 GMT"
```

log

Summary and Conclusion

In Summary, in my role as a backend developer, I have been exposed to technologies like:-

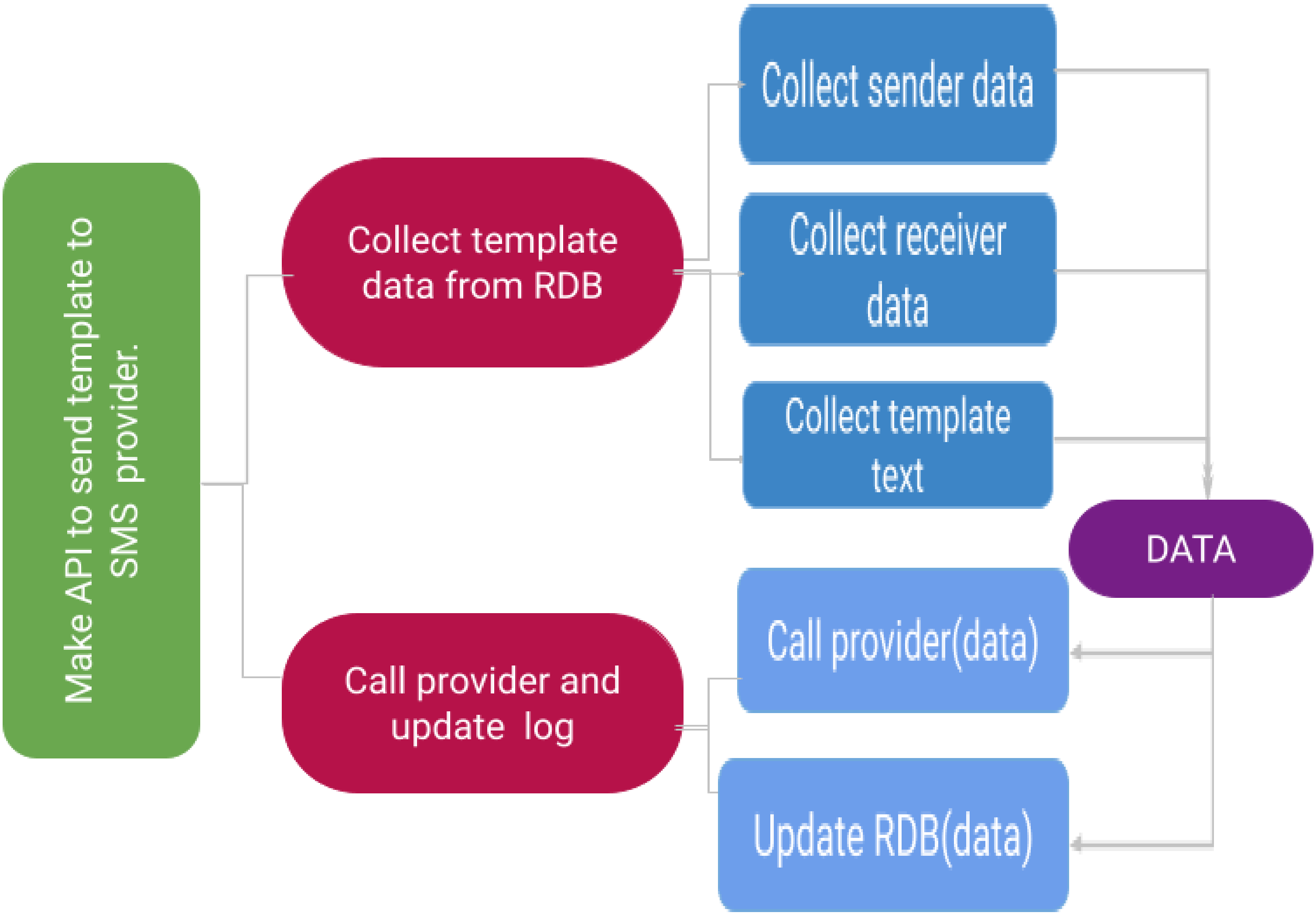
- **Cloud Service models** -IaaS, CPaaS, SaaS.
- Tools used to build **SaaS.**
    - APIs and frameworks - Flask
    - IaaS-AWS services - EC2, RDS, KDS
    - CPaaS- Twilio outbound and webhook service
- Stages of **Production**
    - Relational Database Diagram
    - Prototype Algorithm
    - Testing -on Postman for POST (sms_one) and GET (log)
    - Uploading to the server using SSH clients -Putty and WinSCP
    - Deployment - using Web server Nginx and WSGI server Gunicorn

Post-Mid-Semester

- Deployment - using Web server Nginx and WSGI server Gunicorn
- Sub-features integration -SQS and webhooks
  - Scaling - Amazon SQS
  - Webhooks
  - Google APIs
- Learning Outcome: SDLC
  - Waterfall model
  - Agile model
  - SDLC: Agile vs Waterfall
- Comparing micro frameworks: Flask vs Express on node.js

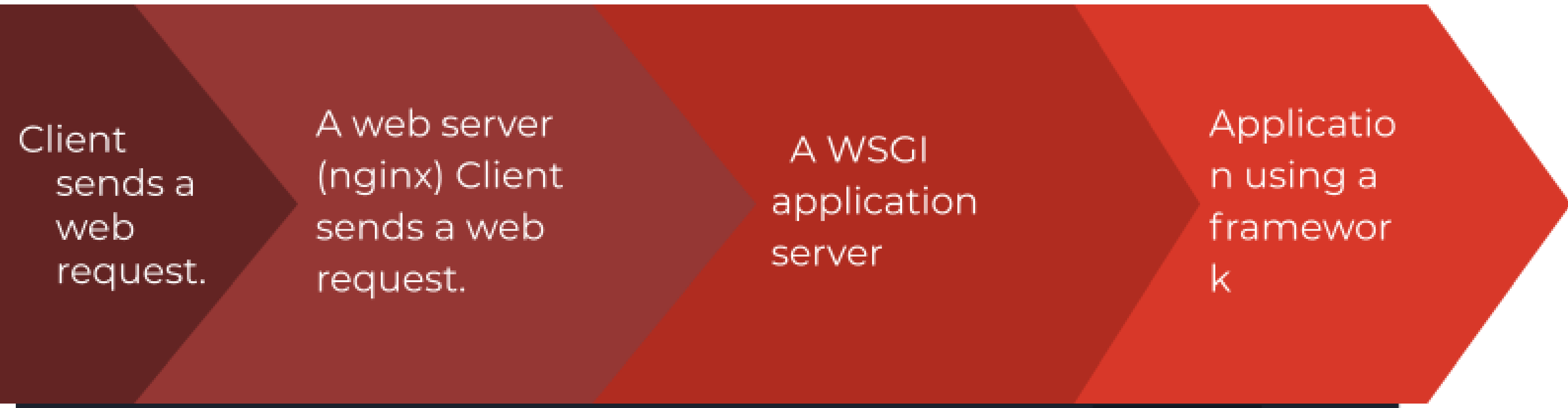
APPENDIX

1. Prototype flowchart



Make API to send template to SMS provider.
Collect template data from RDB
Collect sender data
Collect receiver data
Collect template text
DATA
Call provider and update log
Call provider(data)
Update RDB(data)

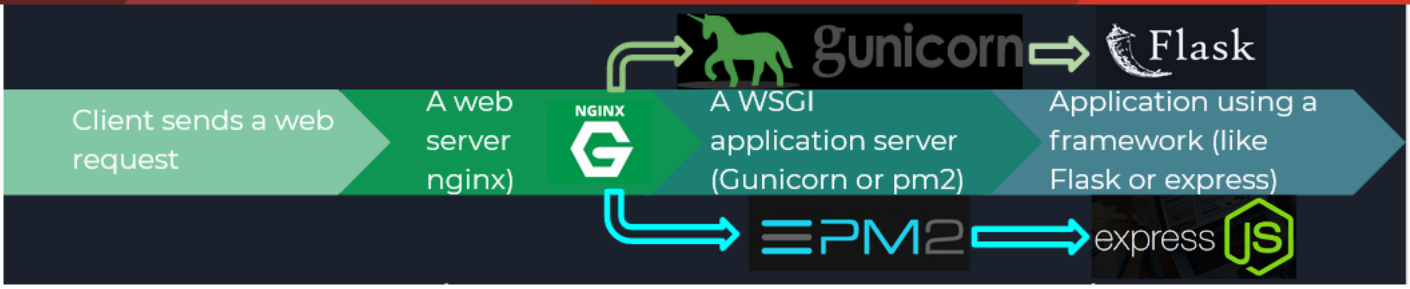
2 Deployment request flow.


Client sends a web request.

A web server (nginx) Client sends a web request.

A WSGI application server

Application using a framework

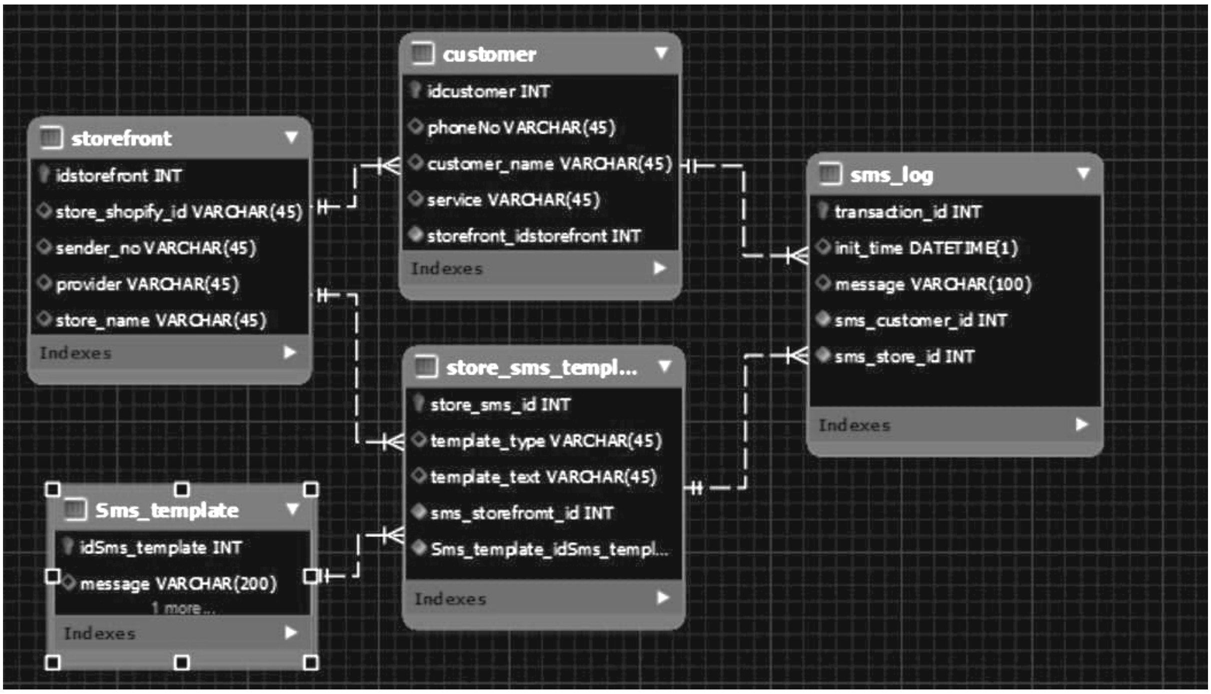

Client sends a web request

A web server nginx

A WSGI application server (Gunicorn or pm2)

Application using a framework (like Flask or express)

4. DATA BASE DIAGRAM USED


storefront
idstorefront INT
store_shopify_id VARCHAR(45)
sender_no VARCHAR(45)
provider VARCHAR(45)
store_name VARCHAR(45)
Indexes

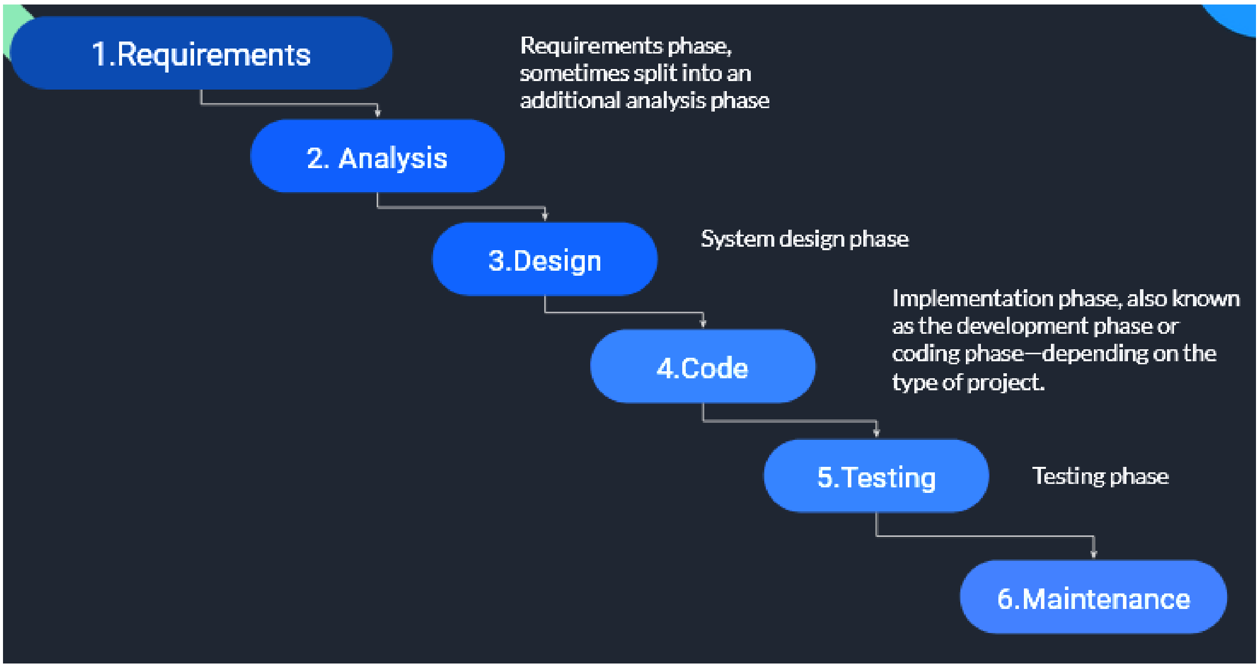
customer
idcustomer INT
phoneNo VARCHAR(45)
customer_name VARCHAR(45)
service VARCHAR(45)
storefront_idstorefront INT
Indexes

sms_log
transaction_id INT
init_time DATETIME(1)
message VARCHAR(100)
sms_customer_id INT
sms_store_id INT
Indexes

store_sms_templ...
store_sms_id INT
template_type VARCHAR(45)
template_text VARCHAR(45)
sms_storefromt_id INT
Sms_template_idSms_templ...
Indexes

Sms_template
idSms_template INT
message VARCHAR(200)
1 more...
Indexes

5. Waterfall Model



1.Requirements

Requirements phase, sometimes split into an additional analysis phase

2. Analysis

3.Design

System design phase

4.Code

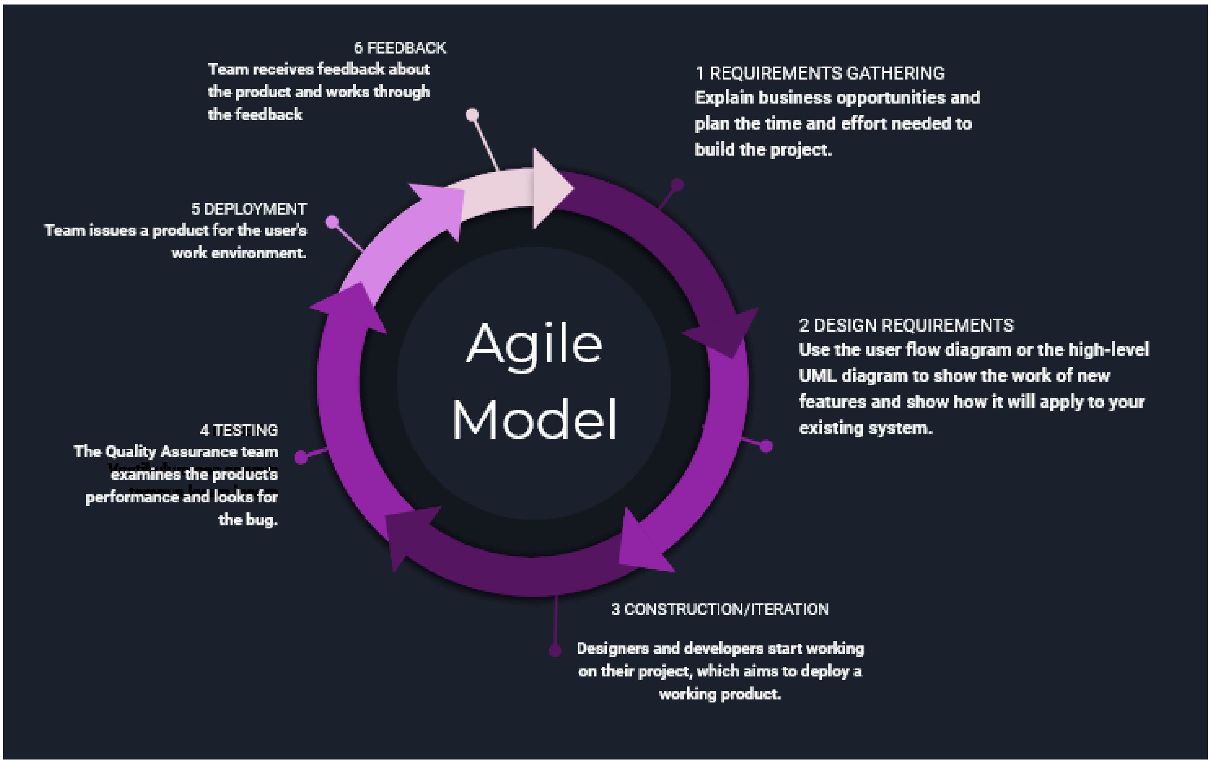
Implementation phase, also known as the development phase or coding phase—depending on the type of project.

5.Testing

Testing phase

6.Maintenance

6. Agile Model



Agile Model

1 REQUIREMENTS GATHERING

Explain business opportunities and plan the time and effort needed to build the project.

2 DESIGN REQUIREMENTS

Use the user flow diagram or the high-level UML diagram to show the work of new features and show how it will apply to your existing system.

3 CONSTRUCTION/ITERATION

Designers and developers start working on their project, which aims to deploy a working product.

4 TESTING

The Quality Assurance team examines the product's performance and looks for the bug.

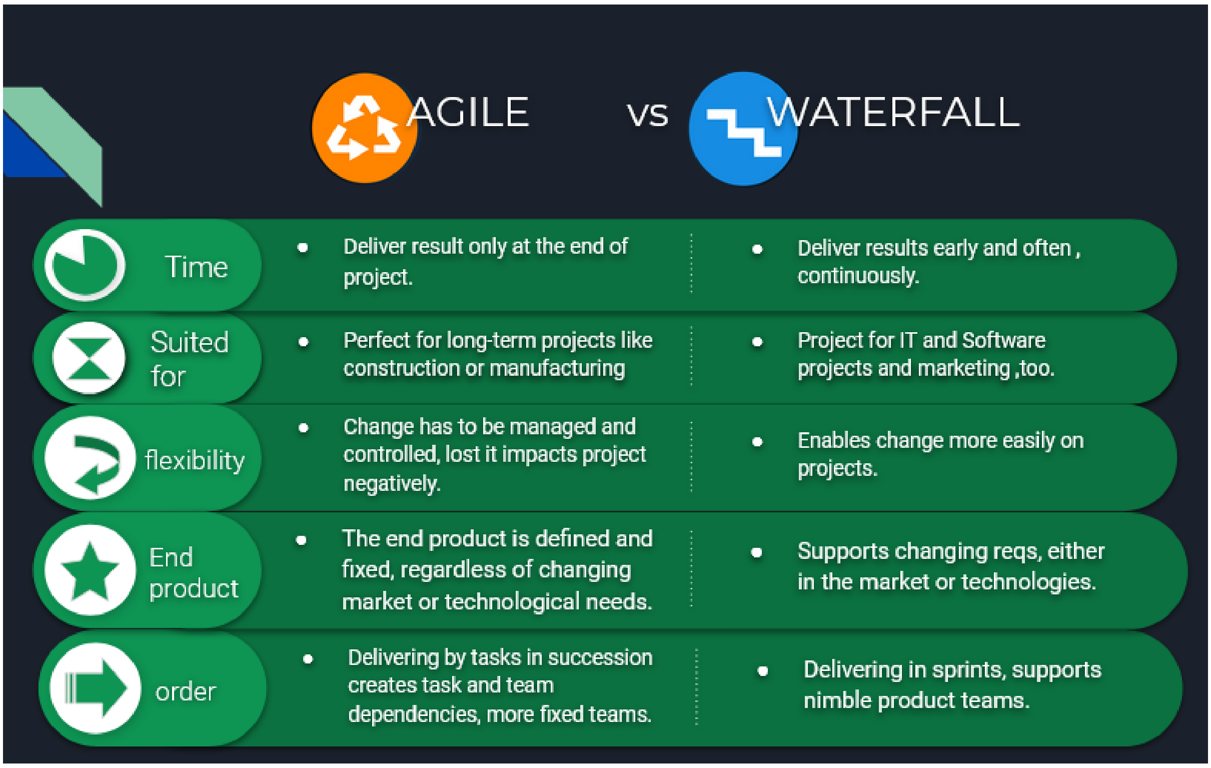
5 DEPLOYMENT

Team issues a product for the user's work environment.

6 FEEDBACK

Team receives feedback about the product and works through the feedback

7. Agile vs waterfall model



	AGILE	vs WATERFALL
Time	Deliver result only at the end of project.	Deliver results early and often , continuously.
Suited for	Perfect for long-term projects like construction or manufacturing	Project for IT and Software projects and marketing ,too.
flexibility	Change has to be managed and controlled, lost it impacts project negatively.	Enables change more easily on projects.
End product	The end product is defined and fixed, regardless of changing market or technological needs.	Supports changing reqs, either in the market or technologies.
order	Delivering by tasks in succession creates task and team dependencies, more fixed teams.	Delivering in sprints, supports nimble product teams.

References

https://joshfechter.com/iaas-paas-saas/

https://www.twilio.com/

https://aws.amazon.com/

https://www.crunchbase.com/organization/mysmartprice

https://www.crunchbase.com/organization/retainiq

https://flask.palletsprojects.com/en/2.0.x/

https://www.nginx.com/resources/glossary/nginx/ttps://build.vsupalov.com/what-is-gunicorn/

https://www.hostinger.com/tutorials/ssh-tutorial-how-does-ssh-work

https://python.plainenglish.io/lets-find-which-is-fast-node-js-vs-flask-python-5ab68b3c3559

https://en.wikipedia.org/wiki/Google_APIs

https://www.innovativearchitects.com/KnowledgeCenter/basic-IT-systes/system-development-life-cycle.aspx

https://www.ibm.com/cloud/blog/agile-vs-waterfall

https://aws.plainenglish.io/deploying-a-nodejs-application-in-aws-ec2-c1618b9b3874