Cook With What You Have!!

- IMT2016022,IMT2016057, IMT2016053, IMT2016078

25th May 2020

Table of Contents

1. Abstract

The purpose of the document is to provide an overall description of the software application and its development using dev-ops tools, as well as it will illustrate the complete declaration for the development of the system. It also explains, system constraints, interface, and interactions with other external applications. This document detail the system's specific requirements, functional and non-functional requirements, database design, standard compliance, system attributes such as reliability, availability, security, maintainability, etc.

2.Introduction

2.1.Importance of Project Problem

At home and feeling hungry, many times we have limited ingredients but don't know what you can make out of them. So our website will come handy in such situations. It will suggest recipes using what ingredients you ALREADY have in your kitchen.

This application will maintain a list of ingredients where you can select them and there will display the list of all possible recipes with those ingredients. Once you select a recipe, all related information regarding that recipe like direction, time, etc will be displayed.

A few conventions that are used in the documentation further are

1. Customer/ User: Person who visits and uses the website.
2. CookThisNotThat Admin: Person who manages the website.
3. Ingredient checkbox: For selecting ingredients among given ingredients.
4. Submit Ingredients: This is a button used for stating you are done with ingredient selection.
5. Recipe checkbox: For selecting a recipe among displayed recipes.
6. Submit recipe: This is a button used for stating you are done with recipe selection.

2.2.Why DevOps approach for this project?

Our project, Cook With What You Have, is very useful in times like this where a pandemic has struck the entire world, and people have to be resourceful and use the food they already have at home. Since this application is of much importance, especially at a time like this, which means that it has to be very fast-paced in its development so that it can be pushed to customers' use in the least possible amount of time.

We clearly know that DevOps is merely only a set of processes that is going to coordinate and unify all the teams and processes working on a software project.

So instead of developers writing code and developing the application step by step, it is more efficient to have teams develop the parts simultaneously and integrate everything at the end. It will also be easier for any kind of bug fixing that needs to be done.
Hence, we chose the DevOps approach to developing this application.

3.System Configuration

OS - Ubuntu version -18.04 ; RAM - 8 GB ; Architecture - x86_64 ; Language used - PHP ; database - Mysql

4.Software Development Life Cycle

4.1.Installation Procedure

- **Installing Maven and JDK**
  - Java 8 can be installed through the following command
    - sudo apt install openjdk-8-jdk
  - Maven can be installed with the following command
    - sudo apt install maven
  - To check their versions and paths use the following commands respectively
    - java -version
    - mvn -version

- **Installing Jenkins**
  - Jenkins is an open-source tool that can automate the software development life cycle. It has plugins which when installed can do continuous integration i.e whenever the code is committed in the GitHub it will automatically build it.
  - Jenkins can be installed in Ubuntu with the following command.
    - sudo apt install jenkins
  - We can start/stop Jenkins with systemctl
    - sudo systemctl start/stop jenkins
  - Once it’s started we can open localhost:8080 in our web browser to see Jenkins page. Give the required details (username, password), install the required plugin and we are good to go.
  - Add JDK 8 and maven paths in the Global tool configuration of manage Jenkins.

- **Installing Docker**
  - Docker is an open-source technology that works like a virtual machine. We can create multiple images and containers to deploy our applications.
  - Docker can be installed through the following command

        - sudo apt install docker-ce
    - The status of it can be checked with systemctl command
        - sudo systemctl status docker

- **Installing Rundeck**
    - Rundeck is an open-source automation service that allows us to add commands in its web console and automates tasks among its nodes.
    - Download the Debian file from the following link http://rundeck.org/download/deb
    - These packages can be installed with the following command
        - sudo dpkg -i <rundeck package name>
    - Rundeck server can be started with the following command
        - sudo service rundeck start
    - If we go to localhost:4440 we can see the login page of the rundeck server with default login details as admin for both username and password.

4.2.Source Code Management (SCM)

- We used Git to manage source code. Git is a Version Control System i.e it manages different versions of our project. We need to create a GitHub account and then we can push our code into it. We can also commit changes to the existing project and roll-back the previously committed changes whenever if necessary. Thus it's an efficient tool to manage large size projects.
- Following are the commands to push a project into GitHub through the command line.
    - Create an empty repository in your GitHub
    - Open terminal in the directory where our code is present. Initialize git there, add all the folders and files present in that directory, commit the changes with a message, add the URL of the empty repository we created in the above step, and finally push the code into GitHub.
        - git init
        - git add .
        - git commit -m “Commit Message”
        - git remote add origin <repository url>
        - git push -u origin master
- The Link to our GitHub Repository is as follows
    - https://github.com/Manonmaie/CookWithWhatYouHave

4.3.Build

The build step basically includes the process that is going to convert files and other assets that a developer produced to a final software product in its consumable form by the consumer. So, in layman terms, build is translating from what a developer delivers to what a customer can use.
In our case, we have created a docker image. For this, we use ‘docker-compose’.

1. docker-compose is usually included in the docker installation, but in case it is not (Linux), first we need to install docker-compose
    - pip install docker-compose
2. We next need to create a Dockerfile that is going to build a docker image. This image will contain all the dependencies(this file can be found in our GitHub repo)
3. Create a docker-compose.yml file in the project directory. This file defines two services: web and db for web and database. It uses the image that is built from the docker file.

Now to build the app using compose

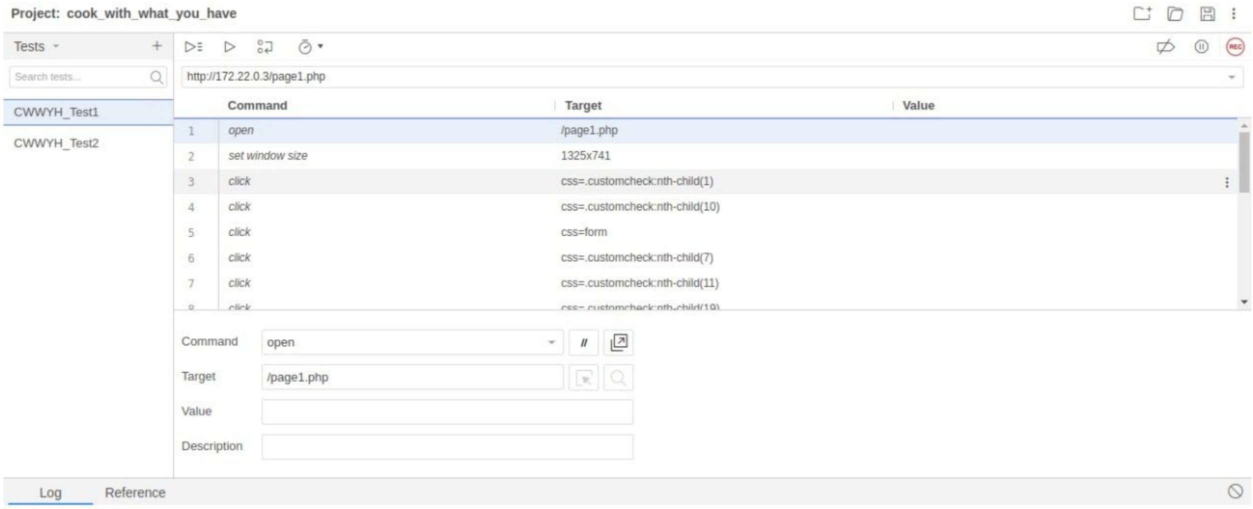
From the project directory, start the application by running

- docker-compose up

It builds an image for our code and starts the services that we defined the docker-compose.yml file. The code is statically copied into the image at build time.
Now confirm if the image has been formed by doing ‘docker images’.
We can check in the localhost port we gave, to see if the application is running.

4.4.Test

- To test our web application we have used selenium.
- Selenium is a portable framework for testing web applications. Selenium provides a playback tool for authoring functional tests without the need to learn a test scripting language.
- While training the selenium we have chosen some ingredients and thereafter chosen a recipe



Project: cook_with_what_you_have

Tests

CWWYH_Test1
CWWYH_Test2

http://172.22.0.3/page1.php

	Command	Target	Value
1	open	/page1.php	
2	set window size	1325x741	
3	click	css=.customcheck:nth-child(1)	
4	click	css=.customcheck:nth-child(10)	
5	click	css=form	
6	click	css=.customcheck:nth-child(7)	
7	click	css=.customcheck:nth-child(11)	
8	click	css=.customcheck:nth-child(19)	

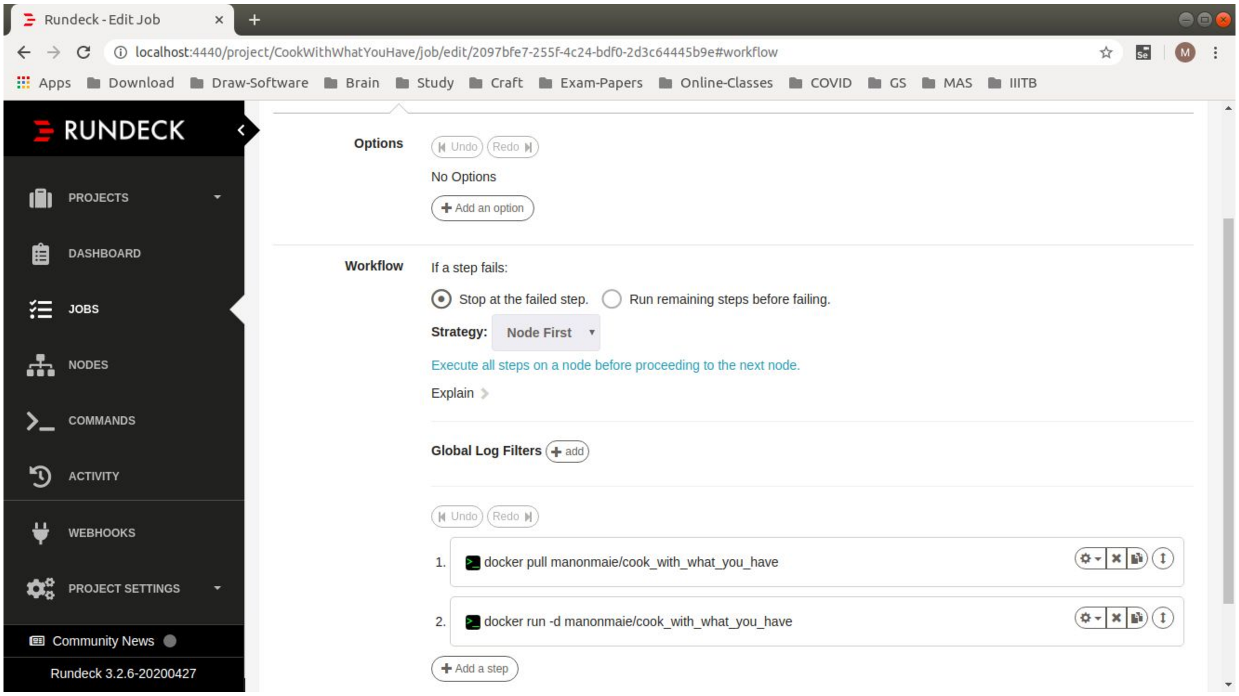
Command: open
Target: /page1.php
Value:
Description:

4.5.Artifact

The Byproduct during the software development process in our project is the docker images created by the docker-compose file. So we had two docker images that needed to be created. One docker for the creation of the SQL database. and other images for the web part. We had to cooperatively run these two images i.e. the containers. We have used docker-compose for this. It takes care of the creation, dependence, and running of the docker images.
Docker Image - docker pull manonmaie/cook_with_what_you_have

4.6.Deploy

- We used Rundek for deploying to the clients. It is the tool that will take an artifact and deploy it to all the nodes that are connected in the configuration. The artifact, here is the docker image will be pulled from DockerHub and the image will be run in the client's system, which results in running the proper application in the client's system, without any configuration from client-side. This will be very helpful for any application deployment, where the developer will be taking whole responsibility for configuration and making the application run client's side.
- There will be a unique id for every job, this id has to be mentioned in the pipeline while integrating.
- Here are the commands that are used in Rundeck configuration for deployment
    1. docker pull <image_name> -- This will pull the required image from DockerHub
    2. docker run -d <image_name> -- To start the required servers and get the application running in the attached nodes



localhost:4440/project/CookWithWhatYouHave/job/edit/2097bfe7-255f-4c24-bdf0-2d3c64445b9e#workflow

Options

No Options

Workflow

If a step fails:

● Stop at the failed step. ○ Run remaining steps before failing.

Strategy: Node First

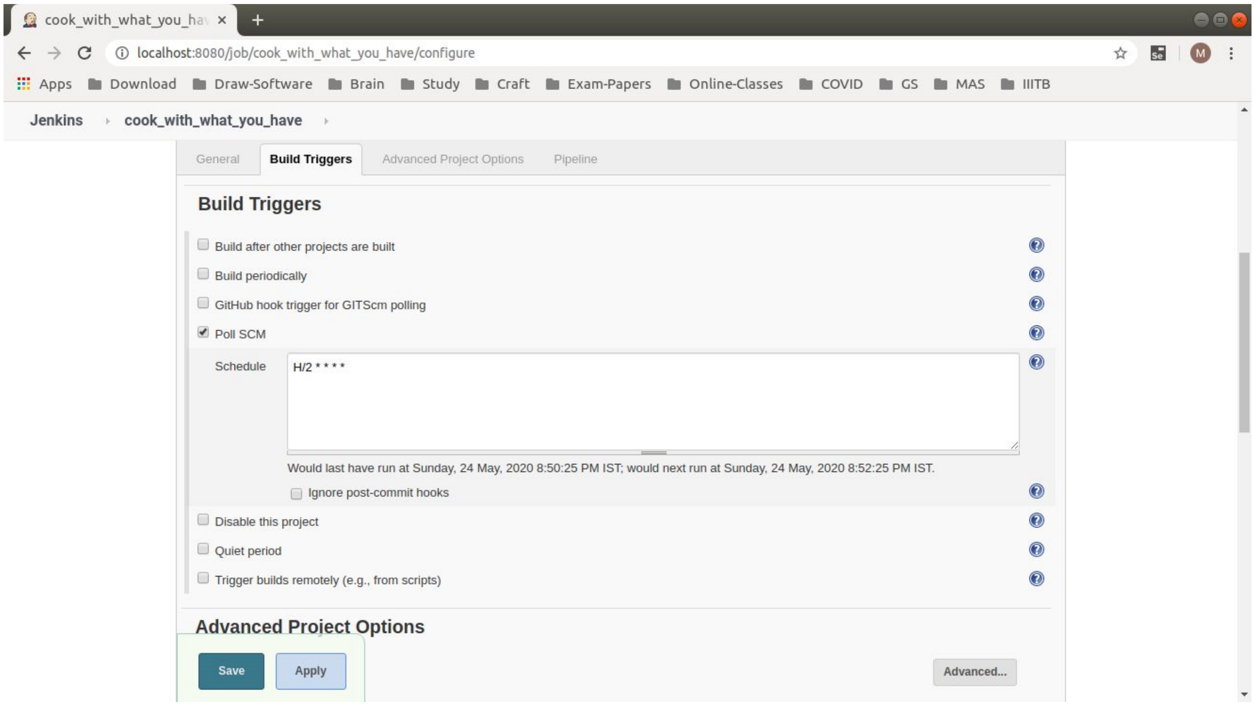
Execute all steps on a node before proceeding to the next node.

Global Log Filters

1. docker pull manonmaie/cook_with_what_you_have
2. docker run -d manonmaie/cook_with_what_you_have

4.7.CI/CD Pipeline

- The tool we used for Continuous Integration is Jenkins. This is the tool that binds all the tools together and will automate the whole process and steps involved in it.
- In Jenkins, we can create a project and configure it to automate the process from the coding stage until the production stage.
- Here we used a Pipeline project so that it'll be easy to monitor all the stages, and this also will make sure that if any intermediate stage fails, the further stages are not run, and thus have the latest working version in the client's systems
- We can configure a project, telling when it should start running the stages, we did it using Poll SCM, which will periodically check the given GitHub Repository for any new commits, if there are any it'll start the stages that are mentioned. Here we are checking every 2 minutes for any new commit



Build Triggers

☐ Build after other projects are built

☐ Build periodically

☐ GitHub hook trigger for GITScm polling

☑ Poll SCM

Schedule: H/2 * * * *

Would last have run at Sunday, 24 May, 2020 8:50:25 PM IST; would next run at Sunday, 24 May, 2020 8:52:25 PM IST.

☐ Ignore post-commit hooks

☐ Disable this project

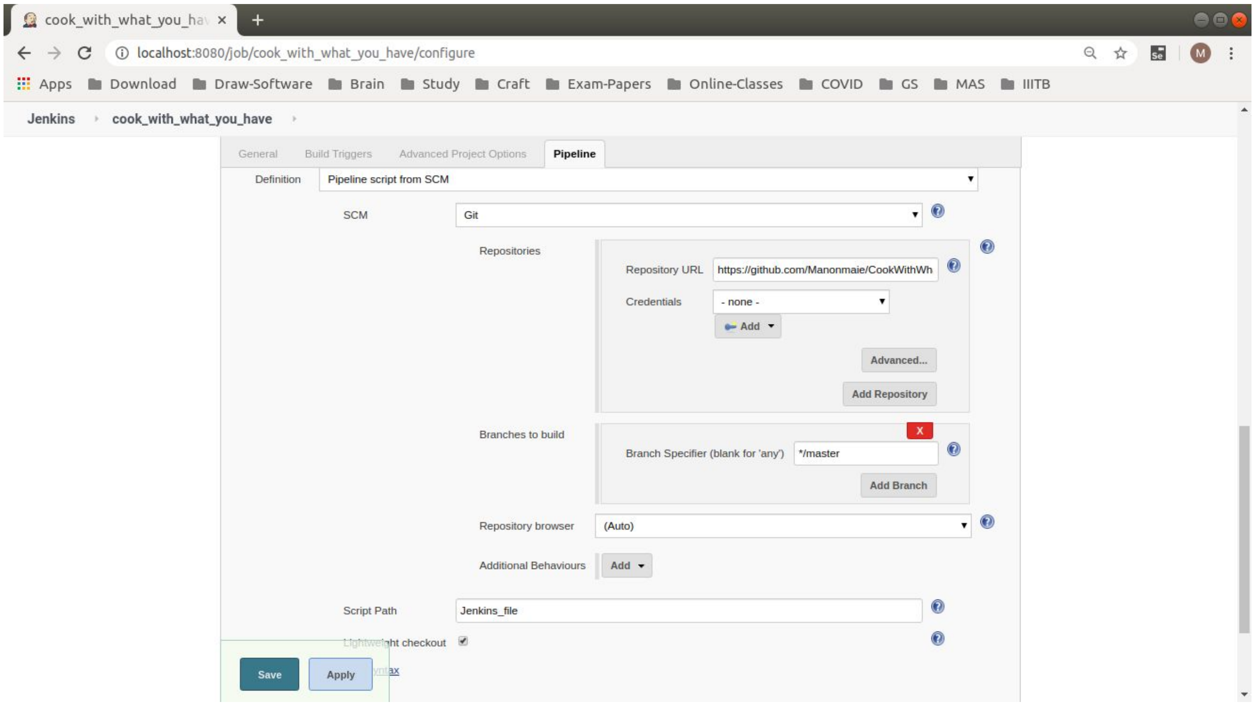
☐ Quiet period

☐ Trigger builds remotely (e.g., from scripts)

Advanced Project Options

- The Stages that are to be triggered and the order is mentioned in the Jenkins file, the configuration for all the stages is also written in Jenkins file
- There are 4 stages in the pipeline created, they are build, run, deliver, and deploy.
  - Each stage is configured using the tools and their configurations that are mentioned above
  - Note that though we did testing using selenium we weren't able to incorporate in the pipeline.
  - The detailed Jenkins file is the git repository that we used for SCM

- The pipeline project that is created is configured by giving the application code i.e., GitHub URL and path to Jenkins file, which contains all the information about each step’s configuration.



Definition: Pipeline script from SCM

SCM: Git

Repositories

Repository URL: https://github.com/Manonmaie/CookWithWh

Credentials: - none -

Branches to build

Branch Specifier (blank for 'any'): */master

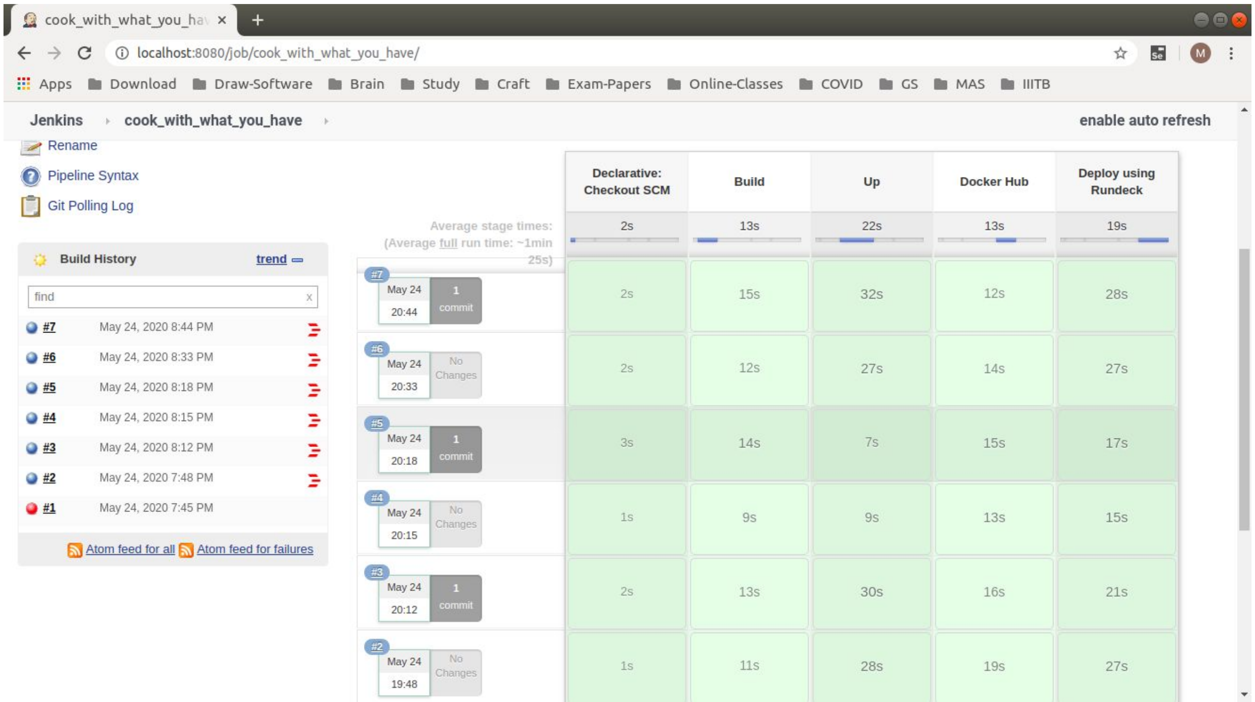
Repository browser: (Auto)

Additional Behaviours

Script Path: Jenkins_file

Lightweight checkout ☑

The pipeline that we have built can be visualized as



Build History

Build	Date
#7	May 24, 2020 8:44 PM
#6	May 24, 2020 8:33 PM
#5	May 24, 2020 8:18 PM
#4	May 24, 2020 8:15 PM
#3	May 24, 2020 8:12 PM
#2	May 24, 2020 7:48 PM
#1	May 24, 2020 7:45 PM

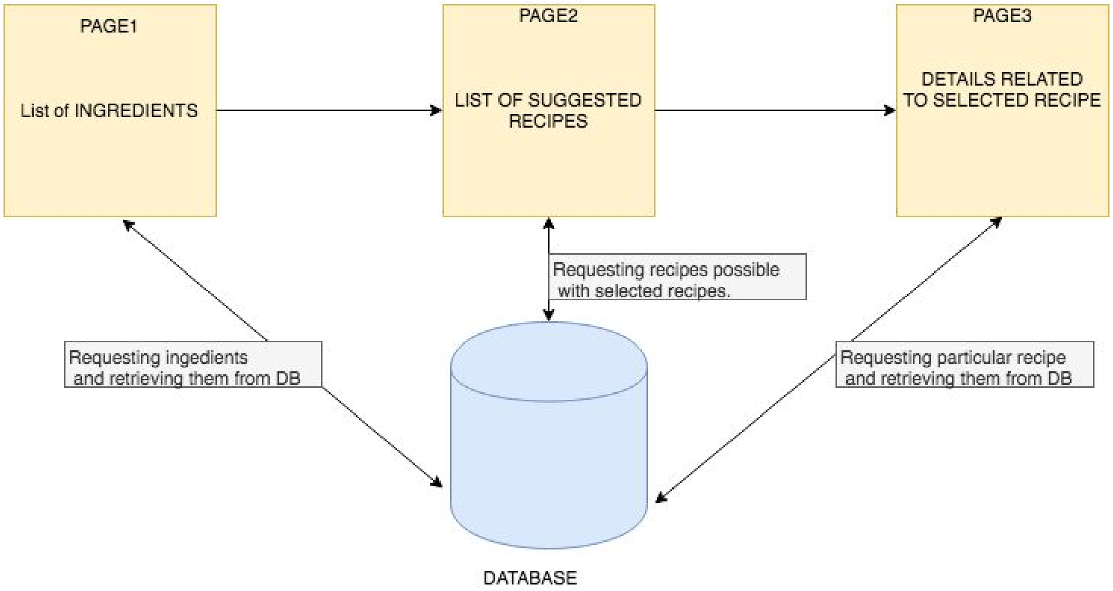
	Declarative: Checkout SCM	Build	Up	Docker Hub	Deploy using Rundeck
Average stage times: (Average full run time: ~1min 25s)	2s	13s	22s	13s	19s
#7 May 24 20:44 1 commit	2s	15s	32s	12s	28s
#6 May 24 20:33 No Changes	2s	12s	27s	14s	27s
#5 May 24 20:18 1 commit	3s	14s	7s	15s	17s
#4 May 24 20:15 No Changes	1s	9s	9s	13s	15s
#3 May 24 20:12 1 commit	2s	13s	30s	16s	21s
#2 May 24 19:48 No Changes	1s	11s	28s	19s	27s

5.Description of the Product

5.1.Product Perspective

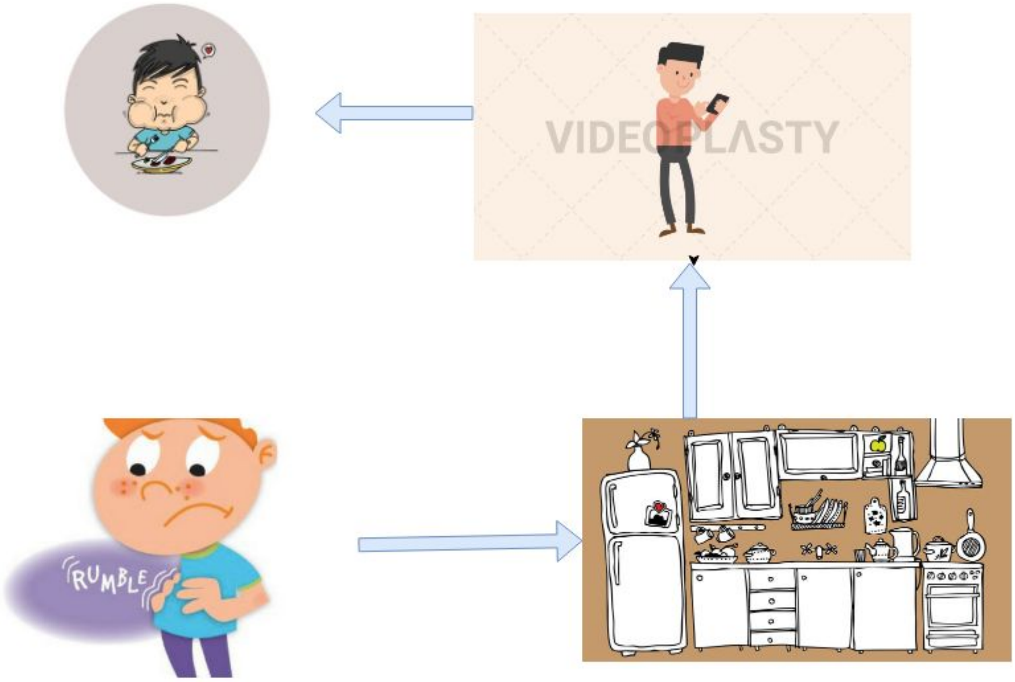
This system is based on a web application. This application consists of the user(customer) interface.

The customer interface consists of all the ingredients to select, once the ingredients are selected it redirects to another page displaying all possible recipes with the selected ingredients. After selecting a specific recipe from those it again redirects to a page showing recipe details.


PAGE1
List of INGREDIENTS
PAGE2
LIST OF SUGGESTED RECIPES
PAGE3
DETAILS RELATED TO SELECTED RECIPE
Requesting recipes possible with selected recipes.
Requesting ingredients and retrieving them from DB
Requesting particular recipe and retrieving them from DB
DATABASE

5.2.Product Functions

- Display Ingredients.
- Select ingredients.
- Display Recipes with selected ingredients.
- Select a particular ingredient.
- Display ingredient details



5.3.User Stories

Users Involved

- General User
- Admin

User Stories of General user

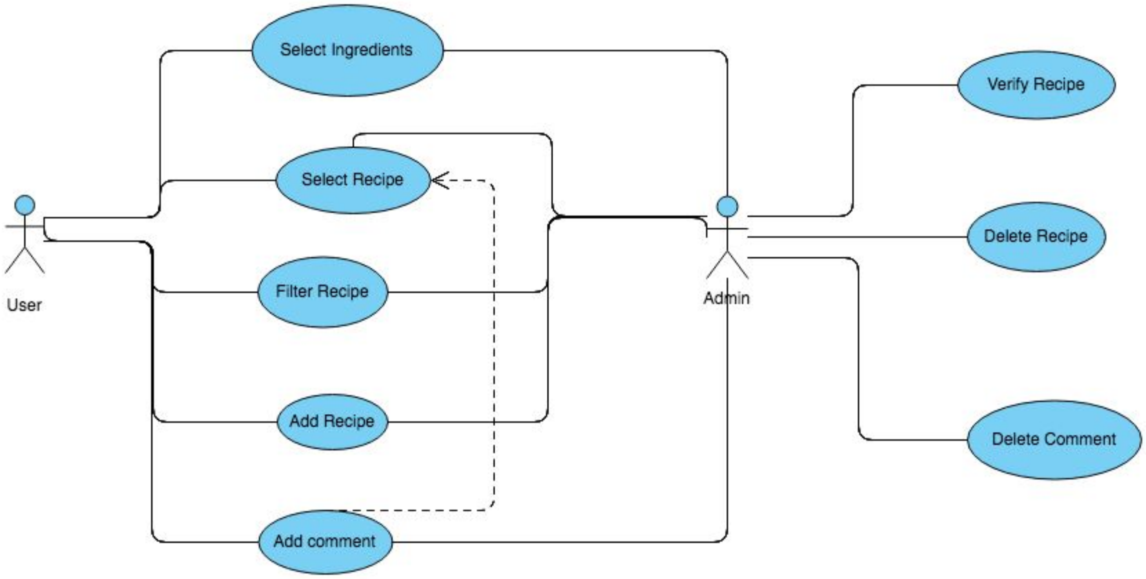
- As a general user, I want to view the general list of ingredients.
- As a general user, I want to select ingredients.
- As a general user, I want to search for a specific ingredient.
- As a general user, I want to see recipes with selected ingredients.
- As a general user, I want to filter recipes based on categories like cooking time, category (like dessert, beverages).

- As a general user, I want to select a particular recipe among all possible recipes which can be prepared from the selected ingredients.
- As a general user, I want to comment on a particular recipe.
- As a general user, I want to see directions for preparation, cooking time, calories.
- As a general user, I want to upload my own recipes.

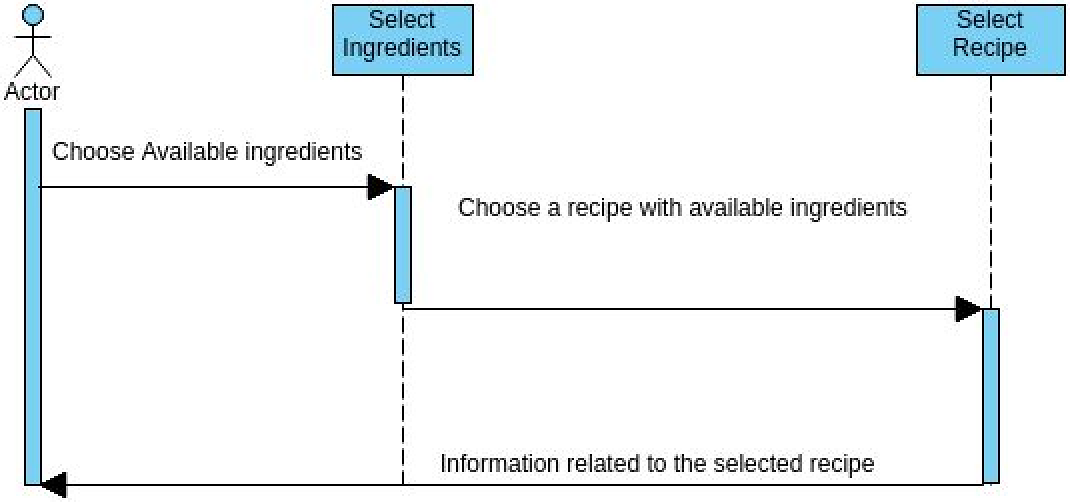
User stories of Admin

- As an admin, I should be able to remove a recipe.
- As an admin, I should be able to monitor the uploaded recipes.
- As an admin, only I should have the authority to add the recipe uploaded by anyone.
- As an admin, I should be able to maintain ingredients.
- As an admin, I should be able to monitor the comments.
- As an admin, I should be able to add a recipe.

5.4.Use Case Diagram



5.5.Sequence Diagram


Actor
Select Ingredients
Select Recipe
Choose Available ingredients
Choose a recipe with available ingredients
Information related to the selected recipe

5.6.Non-Functional Requirements

Performance Requirements

- The product will be based on a local server.
- The performance will depend upon hardware components.

Software Quality Attributes

As the customers are not bound with a particular device or operating system to use the mobile application, adaptability becomes an important requirement. The design should work on multiple smartphone devices such as iOS, Android, Windows, or Symbian with different screen sizes and resolutions.

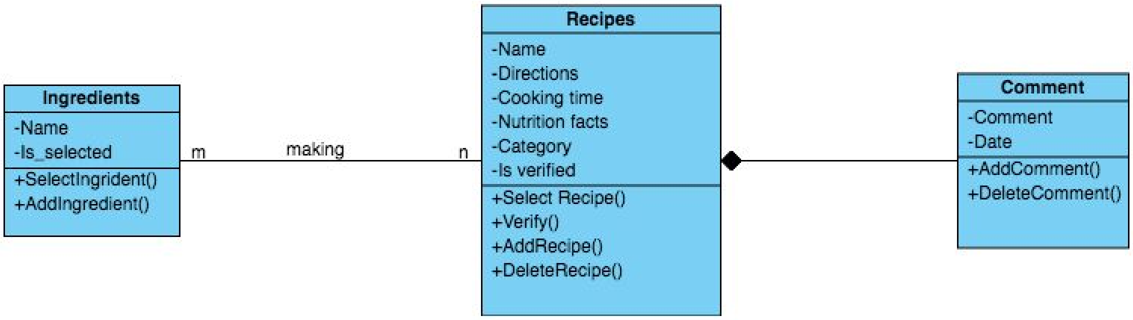
5.7.Operating Environment

Software: A browser with the internet.
This Application is a web-based application. The user should have a computer or a mobile with a browser installed on their devices.

5.8.Design and Implementation Constraints

- Must have a browser to access this web application.
- Memory Requirements: Sufficient memory to manage the browser application.
- Database: Accessed using MySQL language.
- Interface To other Apps: Browser.
- Language Requirements: Knowledge of MySQL, PHP.
- Connectivity: Should be able to connect to the Internet.

5.9.UML Diagram

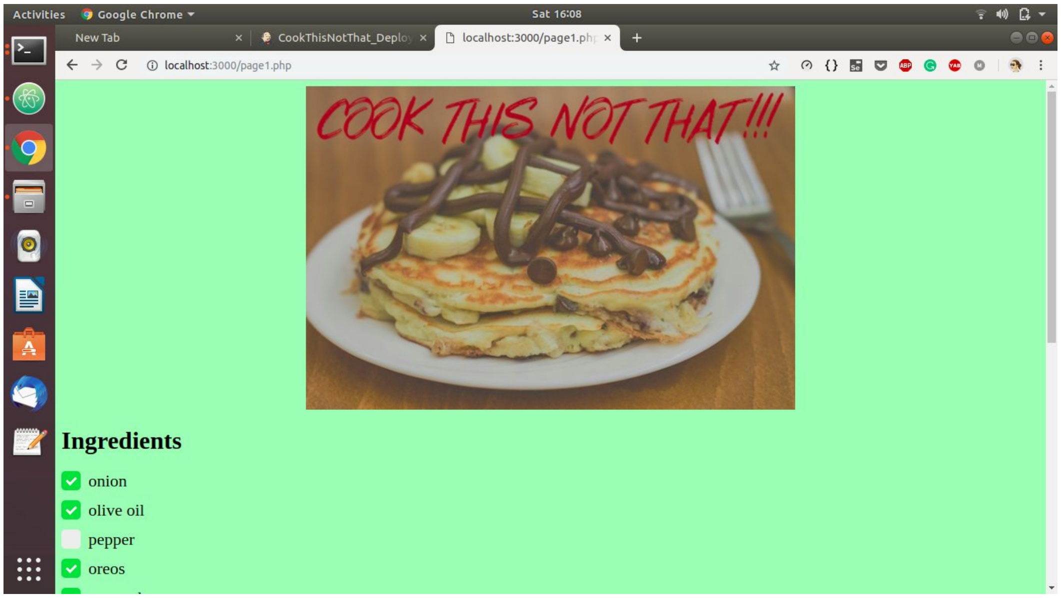


6.Results

6.1.User Interfaces

This interface will be user-friendly.

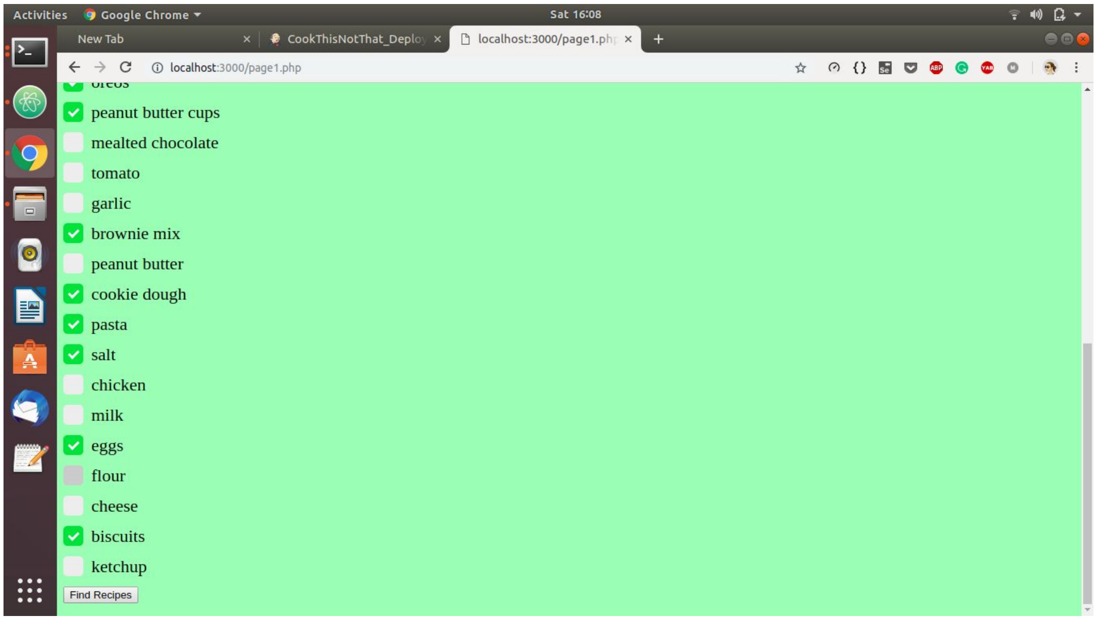
- The display of ingredients will be seen as soon as the website is opened.



Ingredients

☑ onion
☑ olive oil
□ pepper
☑ oreos

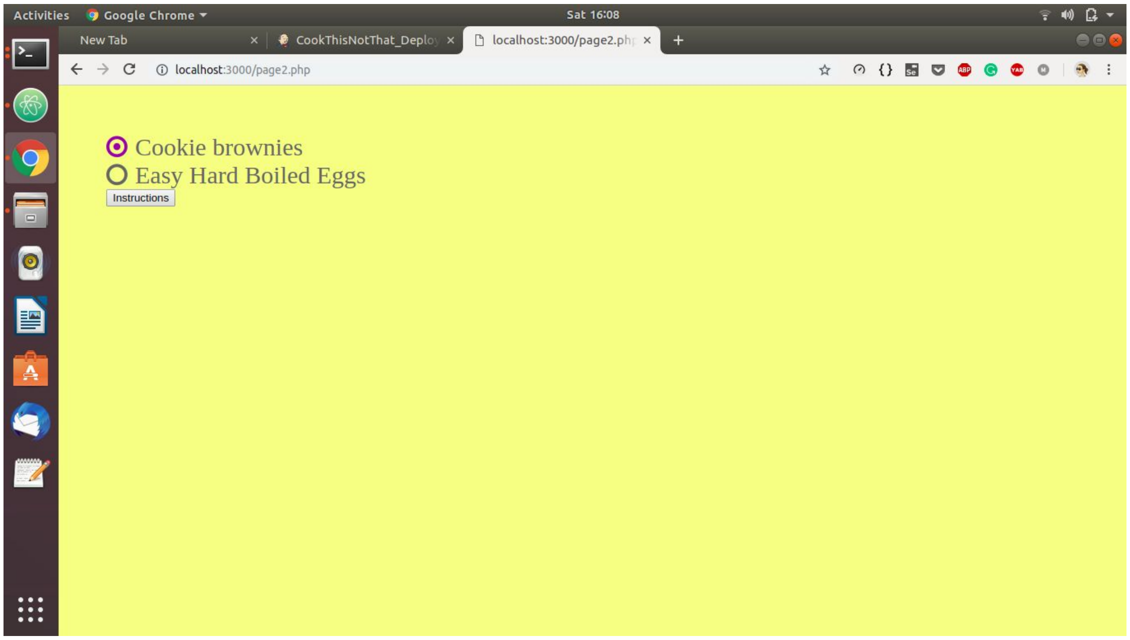
- After selecting the ingredients you can find the recipe by clicking the button at the bottom of the page.



☑ oreos
☑ peanut butter cups
☐ mealted chocolate
☐ tomato
☐ garlic
☑ brownie mix
☐ peanut butter
☑ cookie dough
☑ pasta
☑ salt
☐ chicken
☐ milk
☑ eggs
☐ flour
☐ cheese
☑ biscuits
☐ ketchup

Find Recipes

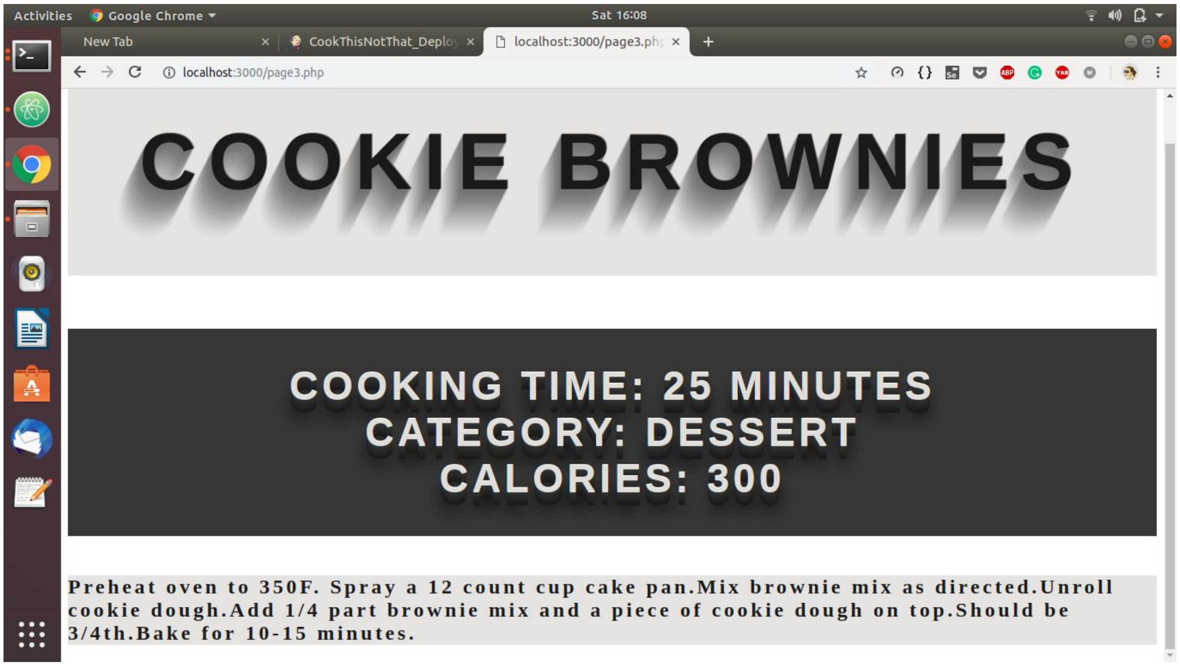
- Page redirected to a page having of suggested recipes



● Cookie brownies
○ Easy Hard Boiled Eggs

Instructions

- Information related to the selected recipe after the selection of a particular recipe.



COOKIE BROWNIES

COOKING TIME: 25 MINUTES
CATEGORY: DESSERT
CALORIES: 300

Preheat oven to 350F. Spray a 12 count cup cake pan.Mix brownie mix as directed.Unroll cookie dough.Add 1/4 part brownie mix and a piece of cookie dough on top.Should be 3/4th.Bake for 10-15 minutes.

7.Future Scope

- Recipe names will be displayed along with their images.
- Many of the times you might not know the name of the ingredient in your kitchen or you might feel lazy to search it in the list of ingredients in our website, so it would be really easy for the user if we could enhance this application where one could scan the ingredient.

8.References

1. https://linuxize.com/post/how-to-install-apache-maven-on-ubuntu-18-04/
2. https://www.digitalocean.com/community/tutorials/how-to-install-java-with-apt-on-ubuntu-18-04
3. https://www.digitalocean.com/community/tutorials/how-to-install-jenkins-on-ubuntu-18-04
4. https://www.edureka.co/blog/what-is-jenkins/
5. https://www.hostinger.in/tutorials/how-to-install-docker-on-ubuntu
6. https://blog.koley.in/2018/install-and-setup-rundeck-on-ubuntu
7. https://codeburst.io/git-and-github-in-a-nutshell-b0a3cc06458f
8. https://help.github.com/en/github/importing-your-projects-to-github/adding-an-existing-project-to-github-using-the-command-line
9. https://foxutech.com/what-is-rundeck/
10. https://www.techrepublic.com/article/how-to-build-a-docker-compose-file/
11. https://github.com/docker/kitematic/issues/2175
12. https://docs.docker.com/compose/reference/run/

13. https://docs.docker.com/docker-hub/publish/publish/
14. https://docs.docker.com/compose/gettingstarted/
15. https://experitest.com/selenium-testing/jenkins-pipeline-selenium/
16. https://medium.com/@sonaldwivedi/jenkins-setup-in-ubuntu-6a7ee12b648f