A
Project Report
On
"E-Store"

Prepared by

PRITESH VANDRA (20DIT102)
PARTH VATALIYA (20DIT103)

Under the guidance of

Prof. Akash Patel

A Report Submitted to

Charotar University of Science and Technology
for Partial Fulfillment of the Requirements for the
5th Semester Software Group Project-III (IT353)

Submitted at

INFORMATION TECHNOLOGY

DEVANG PATEL INSTITUTE OF RESEARCH AND TECHNOLOGY

At: Changa, Dist: Anand – 388421

NOVEMBER 2022

CHARUSAT
CHAROTAR UNIVERSITY OF SCIENCE AND TECHNOLOGY

CERTIFICATE

This is to certify that the report entitled "**E-Store**" is a bonafied work carried out by **Mr. Pritesh M. Vandra (20DIT102)** under the guidance and supervision of **Prof. Akash Patel** for the subject **IT353 -Software Group Project-III** of 5th Semester of Bachelor of Information Technology in **Devang Patel Institute Of Research And Technology** at Faculty of Technology & Engineering – CHARUSAT, Gujarat.

To the best of my knowledge and belief, this work embodies the work of candidate himself, has duly been completed, and fulfills the requirement of the ordinance relating to the B.Tech. Degree of the University and is up to the standard in respect of content, presentation and language for being referred to the examiner.

Prof. Akash Patel
Department of Information and Technology
DEPSTAR, Changa, Gujarat.

Dr. Amit Nayak,
Head of Department(IT),
DEPSTAR,
CHARUSAT, Changa, Gujarat

Dr. Y. P. Kosta
I/C Principal,
DEPSTAR,
CHARUSAT, Changa, Gujarat.

Devang Patel Institute of Advance Technology And Research At: Changa, Ta. Petlad, Dist. Anand, PIN: 388 421. Gujarat

CHARUSAT
CHAROTAR UNIVERSITY OF SCIENCE AND TECHNOLOGY

CERTIFICATE

This is to certify that the report entitled "**E-Store**" is a bonafied work carried out by **Mr. Parth P. Vataliya (20DIT103)** under the guidance and supervision of **Prof. Akash Patel** for the subject **IT353 -Software Group Project-III** of 5th Semester of Bachelor of Information Technology in **Devang Patel Institute Of Research And Technology** at Faculty of Technology & Engineering – CHARUSAT, Gujarat.

To the best of my knowledge and belief, this work embodies the work of candidate himself, has duly been completed, and fulfills the requirement of the ordinance relating to the B.Tech. Degree of the University and is up to the standard in respect of content, presentation and language for being referred to the examiner.

Prof. Akash Patel
Department of Information and Technology
DEPSTAR, Changa, Gujarat.

Dr. Amit Nayak,
Head of Department(IT),
DEPSTAR,
CHARUSAT, Changa, Gujarat

Dr. Y. P. Kosta
I/C Principal,
DEPSTAR,
CHARUSAT, Changa, Gujarat.

Devang Patel Institute of Advance Technology And Research At: Changa, Ta. Petlad, Dist. Anand, PIN: 388 421. Gujarat

DECLARATION BY THE CANDIDATE

I hereby declare that the project report entitled "**E-Store**" submitted by me to Devang Patel Institute of Advance Technology and Research, Changa in partial fulfilment of the requirement for the award of the degree of **B.Tech** in Information Technology, from Institute of Advance Technology and Research, is a record of Bonafede IT353 Software Project Major (project work) carried out by me under the guidance of **Prof. Akash Patel.** I further declare that the work carried out and documented in this project report has not been submitted anywhere else either in part or in full and it is the original work, for the award of any other degree or diploma in this institute or any other institute or university.

Pritesh Vandra (20DIT102)

Parth Vataliya (20DIT103)

Prof. Akash Patel

Teaching cum Research Assistant

Devang Patel Institute of Advance Technology and Research,

DEPSTAR, CHARUSAT – Changa

Abstract

Agile practices with Continuous Integration and Continuous Delivery (CICD) pipeline approach has increased the efficiency of projects. In agile, new features are introduced to the system in each sprint delivery, and although it may be well developed, the delivery failures are possible due to performance issues. By considering delivery timeline, moving for system scaling is common solution in such situations.

The traditional load test methods are unable to identify production performance behavior due to simulated traffic patterns are highly deviated from production. To overcome those issues, this approach has extended CICD pipeline to have three automation phases named benchmark, load test and scaling.

It minimizes the system interruption by using test bench approach when system benchmarking and it uses the production traffic for load testing which gives more accurate results. Once benchmark and load test phases are completed, system scaling can be evaluated.

Initially, the pipeline was used to developed using Jenkins CI server, Git repository and Nexus repository with Ansible automation. Then GoReplay is used for traffic duplication from production to test bench environment. Nagios monitoring is used to analyze the system behavior in each phase and the result of test bench has proven that scaling is capable to handle the same load while changing the application software, but it doesn't optimize response time of application at significant level and it helps to reduce the risk of application deployment by integrating this three phase approach as CICD automation extended feature. Thereby the research provides effective way to manage Agile based CICD projects.

ACKNOWLEDGEMENT

We, as the developer of a deployed e-commerce application “E-Store”, with immense pleasure and commitment would like to present the project assignment. The development of this project has given me wide opportunity to think, implement and interact with various aspects of management skills as well as the new emerging technologies.

Every work that one completes successfully stands on the constant encouragement, good will and support of the people around. We hereby avail this opportunity to express my gratitude to number of people who extended their valuable time, full support and cooperation in developing the project.

I express deep sense of gratitude towards our Head of the IT Department, Dr. Amit Nayak and project guides Prof. Akash Patel for the support during the whole session of study and development. It is because of them, that We were prompted to do hard work, adopting new technologies.

They provided us favourable environment, and without them it would not have been possible to achieve my goal.

TABLE OF CONTENTS

Chapter 1 Introduction

Chapter 2 Project Management

Chapter 3 System Requirements Study

Chapter 4 System Functionality

Chapter 5 Screenshots and limitations

Chapter 6 Outcomes and Future enhancements

Chapter 7 Conclusion

LIST OF FIGURES

CHAPTER-1 INTRODUCTION

1.1 Project Overview

Any online business, requires features such as wishlist, products categorization, various types of filters, payments, customer management and backend. E-commerce app is designed to satisfy all these requirements so that it saves the valuable time of our customers and gives a wide variety of options to choose from at one stop.

The interface is designed using flutter, a multi-platform supporting framework. Also, this application involves a lot of data transfer, as a user is expected to scroll through a lot of products before purchasing.

Firebase, backed for backend, makes it a faster and smoother experience for the users. The primary goal of e-commerce is to reach maximum customers at the right time to increase sales and profitability of the business.

1.2 Scope

E-commerce is quickly becoming a recognised and widely utilised business model. More and more companies are giving capabilities for conducting commercial transactions using applications.

It is fair to state that the procedure of purchasing on apps is getting more frequent. With the introduction of these services, many new start-ups may now readily offer their products and services online.

The majority of small company owners utilise web interfaces to take their operations online. Using a cross-platform app will help them reach a larger percentage of returning clients. The present small-scale internet retailers lack a user-friendly interface. There are several limits to the existing system, however solutions can be supplied in the future.

1.3 Tools and Technologies

Flutter :

This app is created with Flutter. Flutter is the latest framework in the world of mobile app development. Digging deeper here What is the Flutter framework, its strengths and weaknesses, and different ways to test Flutter application. The Flutter framework consists of both a software development kit (SDK) and its widgetbased kit. Flutter does not use native UI components. It may sound unpleasant at first. However, since the components are implemented in Flutter itself, there is no communication layer in between.

Firebase :

The Firebase Framework that's used on this software is useful for constructing one of these database. To replace the records and those adjustments are contemplated right away to each consumer. Firebase carrier handles maximum of the server-aspect paintings and there are numerous different factors that makes firebase a becoming preference as a unfastened BaaS for applications. These offerings encompass a realtime database that's hosted on cloud in which the records is saved as JSON files, it additionally permits the builders to authenticate the consumer the use of emails and passwords, additionally firebase affords garage for all of the records this is generated through the consumer, which in our software is the approximately segment and all of the records entered throughout the registration of events.

Jenkins :

Jenkins offers a simple way to set up a continuous integration or continuous delivery (CI/CD) environment for almost any combination of languages and source code repositories using pipelines, as well as automating other routine development tasks. While Jenkins doesn’t eliminate the need to create scripts for individual steps, it does give you a faster and more robust way to integrate your entire chain of build, test, and deployment tools than you can easily build yourself.

CHAPTER-2 PROJECT MANAGEMENT

2.1 Project Planning

Starting of July, we started planning for choosing a project for Software Group Project (SGP). While choosing the project, criteria was the project, which would give us some knowledge, which would have some real problem statement and it should be useful for most of people.

After choosing project **CI/CD pipeline,** we started learning ubuntu and AWS server which were been used in this project. We watched many videos and read many documents related to this but we have the thinking that we should try some different thing that's why we have merged many modules instead of taking one in use.

2.2 Gantt Chat

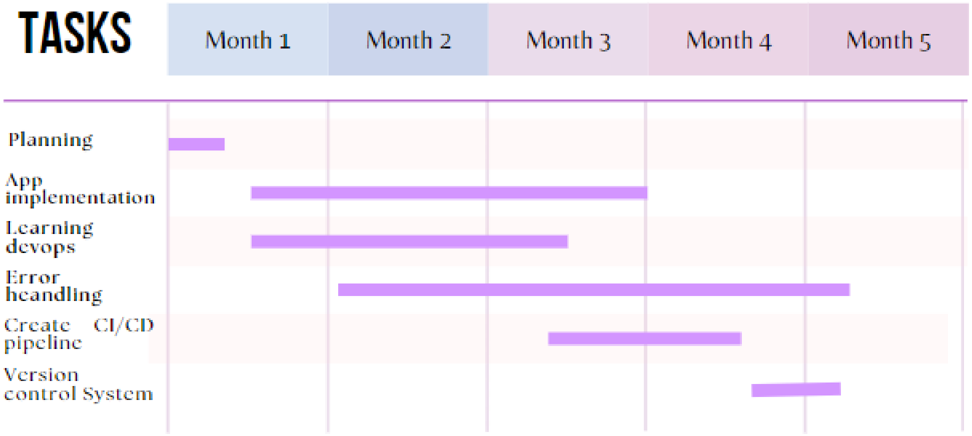


FIGURE-2.1.1 Planning

CHAPTER-3 SYSTEM REQUIREMENTS STUDY

3.1 Hardware Requirements

- Laptop / PC
- Memory of 50 MB RAM (Minimum)
- Processor - All new Generation Processor
- Operating System: Windows 10 (Home) or MacOS
- Storage of 64 GB (Minimum)

3.2 Software Requirements

- GitHub
- Jenkins
- Flutter
- Firebase
- AWS server

CHAPTER-4 SYSTEM FUNCTIONALITY

4.1 Major Functionality :

It is a devops based project in which the user deploy their application, with the help of devops technology which include the Jenkins tool and therefore it will push it into the main server.

4.1.1 Jenkins tool

Jenkins is an open source automation server written in Java. It is used to continuously build and test software projects, enabling developers to set up a CI/CD environment. It also supports version control tools like Subversion, Git, Mercurial, and Maven.

Jenkins is well-documented and extremely extensible, with a rich ecosystem of plugins and integrations. For that reason, it excels when used in large projects that require lots of customization.

While Jenkins itself is free, it must be run on a server which will need attention, updates, and maintenance.

4.1.2 Github

Github is a web-based interface that uses Git, the open source version control software that lets multiple people make separate changes to web pages at the same time. As Carpenter notes, because it allows for real-time collaboration, GitHub encourages teams to work together to build and edit their site content.

4.1.3 Linux Server

A Linux server is a server built on the Linux open-source operating system. It offers businesses a low-cost option for delivering content, apps and services to their clients. Because Linux is open-source, users also benefit from a strong community of resources and advocates.

4.2 Flowchart:

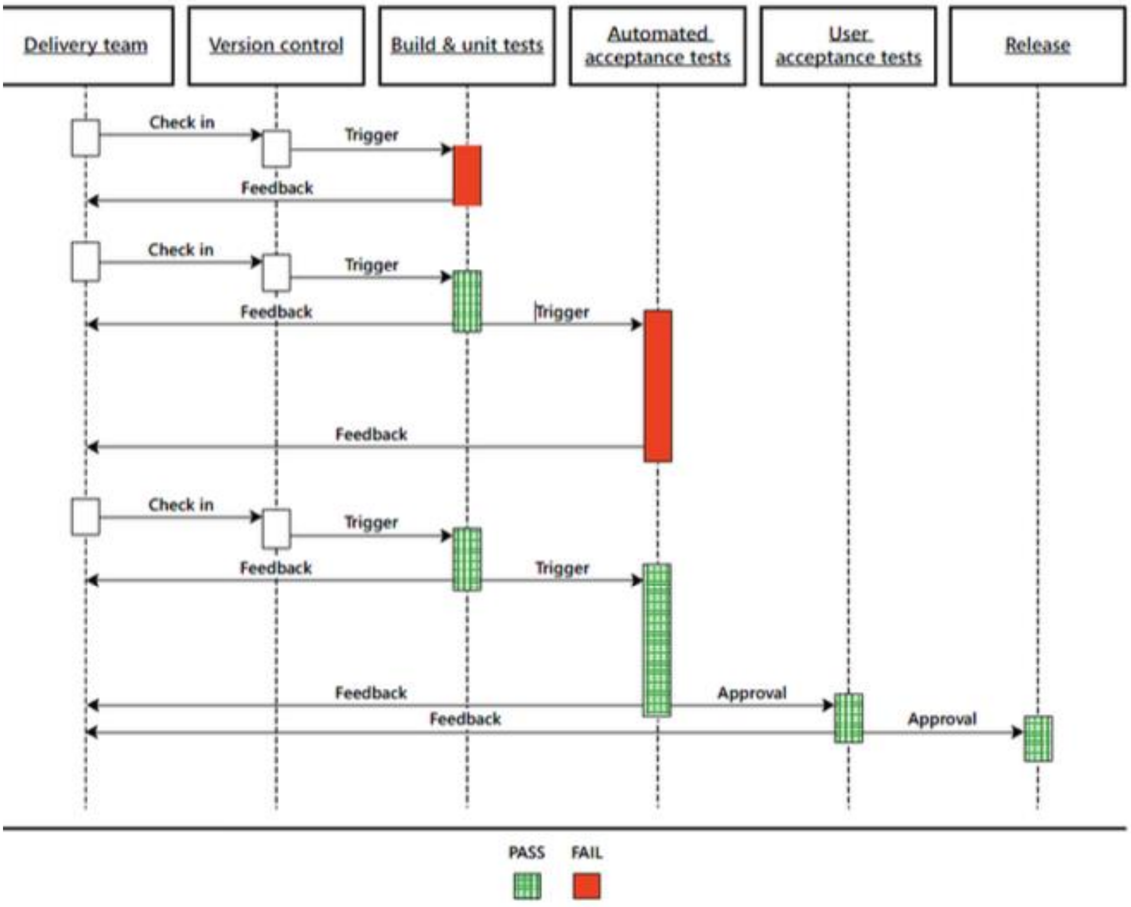


FIGURE-4.2.1 Flowchart

4.3 Implementation Details Of Project:

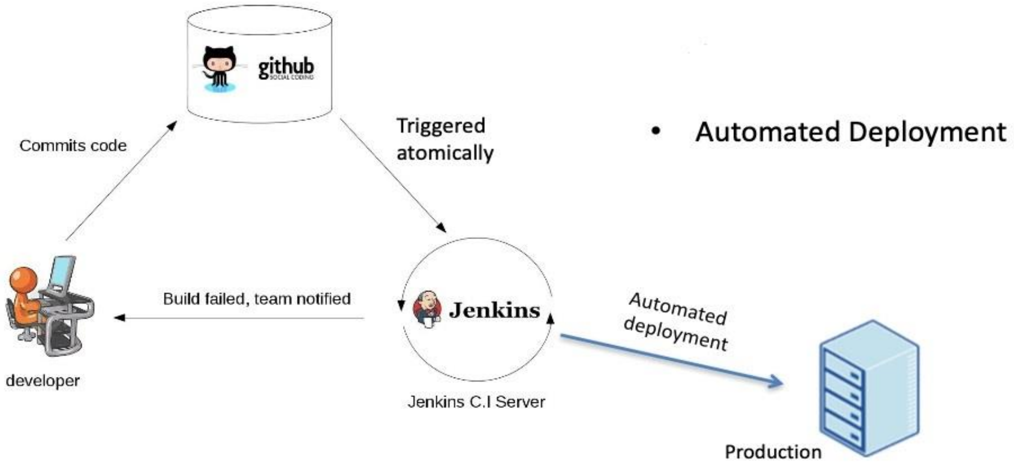


FIGURE-4.3.1 Implementation detail of project

A CI/CD pipeline may sound like overhead, but it isn't. It's essentially a runnable specification of the steps that any developer needs to perform to deliver a new version of a software product. In the absence of an automated pipeline, engineers would still need to perform these steps manually, and hence far less productively.

Most software releases go through a couple of typical stages:

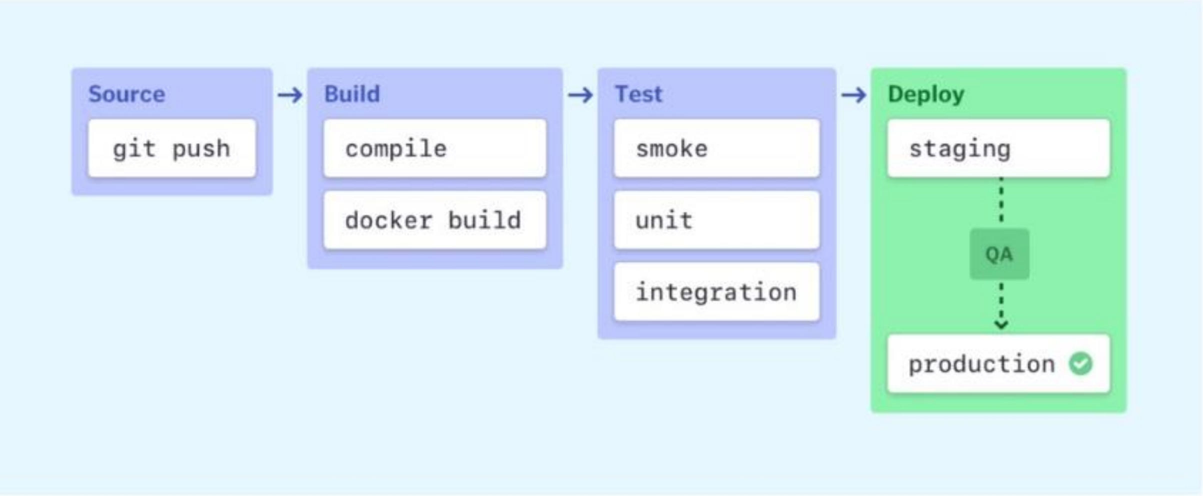


FIGURE-4.3. 2 Stages of CI/CD pipeline

Source stage

In most cases, a pipeline run is triggered by a source code repository. A change in code triggers a notification to the CI/CD tool, which runs the corresponding pipeline. Other common triggers include automatically scheduled or user-initiated workflows, as well as results of other pipelines.

Build stage

We combine the source code and its dependencies to build a runnable instance of our product that we can potentially ship to our end users. Programs written in languages such as Java, C/C++, or Go need to be compiled, whereas Ruby, Python and JavaScript programs work without this step.

Regardless of the language, cloud-native software is typically deployed with Docker, in which case this stage of the CI/CD pipeline builds the Docker containers.

Failure to pass the build stage is an indicator of a fundamental problem in a project's configuration, and it's best to address it immediately.

Test stage

In this phase, we run automated tests to validate our code's correctness and the behavior of our product. The test stage acts as a safety net that prevents easily reproducible bugs from reaching the end-users.

The responsibility of writing tests falls on the developers. The best way to write automated tests is to do so as we write new code in test- or behavior-driven development.

Depending on the size and complexity of the project, this phase can last from seconds to hours. Many large-scale projects run tests in multiple stages, starting with smoke tests that perform quick sanity checks to end-to-end integration tests that test the entire system from the user's point of view. An extensive test suite is typically parallelized to reduce run time.

Failure during the test stage exposes problems in code that developers didn't foresee when writing the code. It's essential for this stage to produce feedback to developers quickly, while the problem space is still fresh in their minds and they can maintain the state of flow.

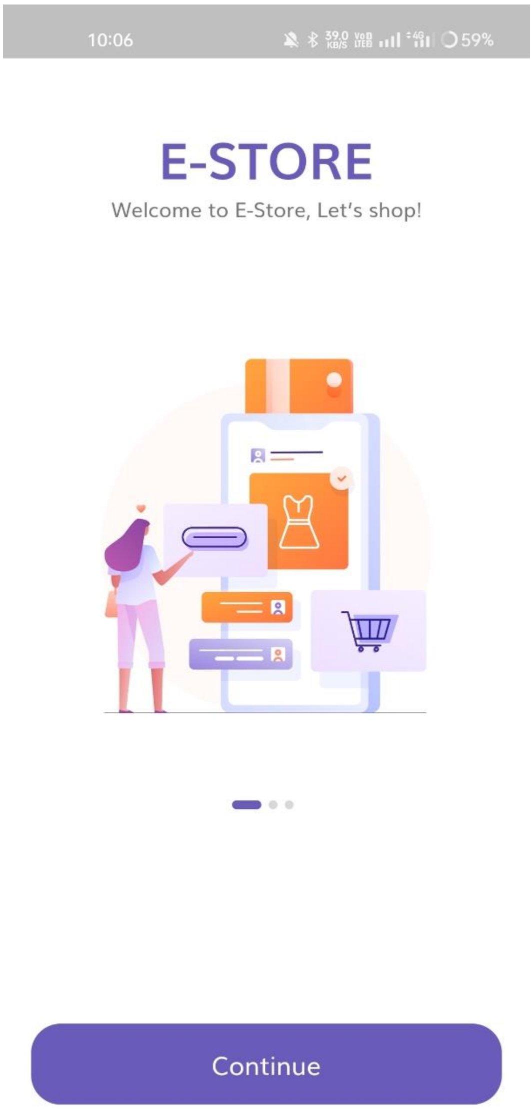
Deploy stages

Once we have a built a runnable instance of our code that has passed all predefined tests, we're ready to deploy it. There are usually multiple deploy environments, for example, a "beta" or "staging" environment which is used internally by the product team, and a "production" environment for end-users.

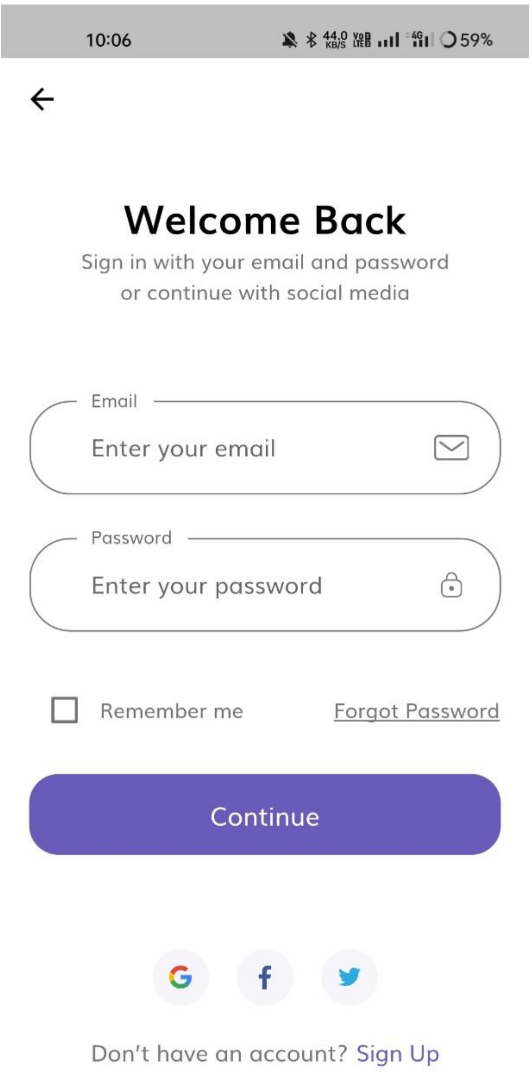
Teams that have embraced the Agile model of development—guided by tests and real-time monitoring—usually deploy work-in-progress manually to a staging environment for additional manual testing and review, and automatically deploy approved changes from the master branch to production.

CHAPTER-5 SCREENSHOTS AND LIMITATIONS

5.1 Screenshots



Welcome to E-Store, Let's shop!



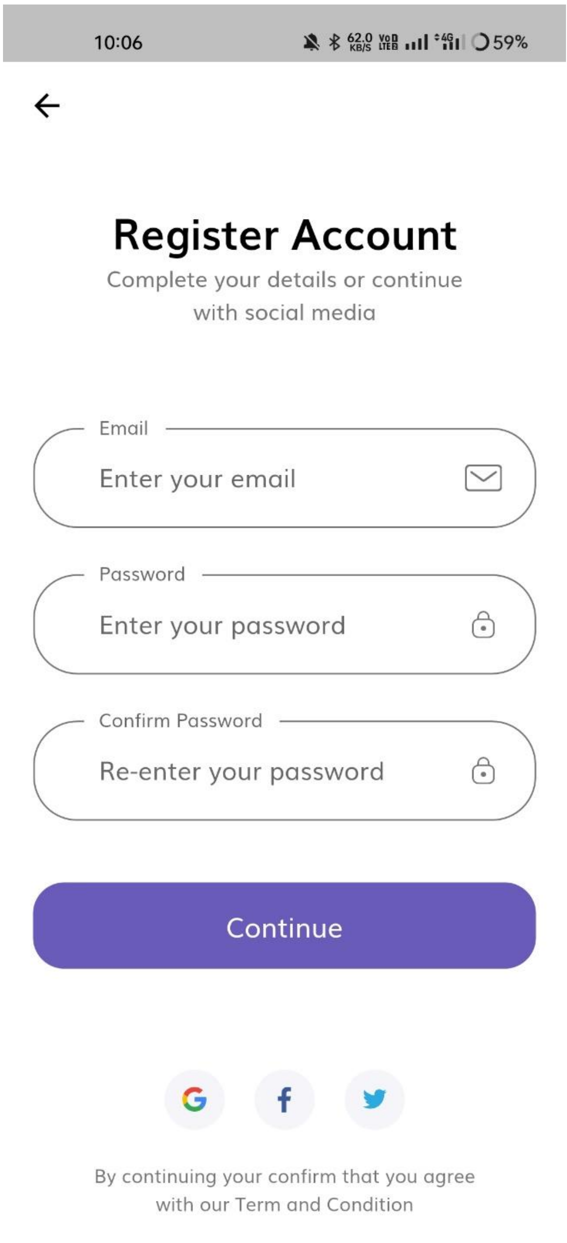
Welcome Back

Sign in with your email and password or continue with social media

Don't have an account? Sign Up

FIGURE-5.1.1 Splash Screen

FIGURE-5.1.2 Sign in Screen



Complete your details or continue with social media

By continuing your confirm that you agree with our Term and Condition

FIGURE-5.1.3 Sign up Screen

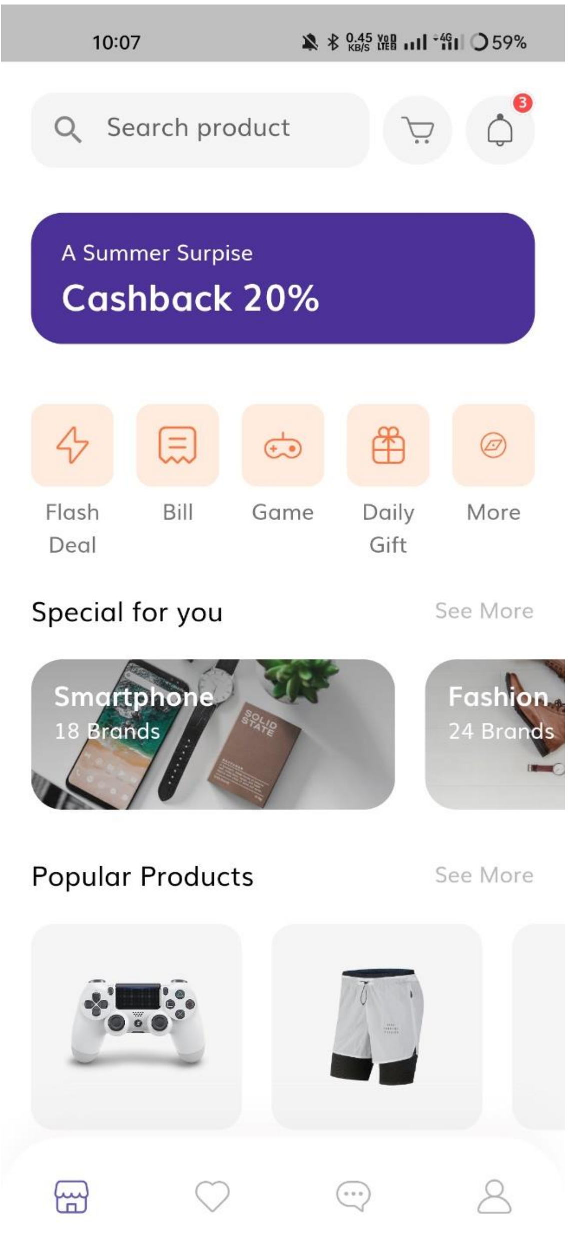
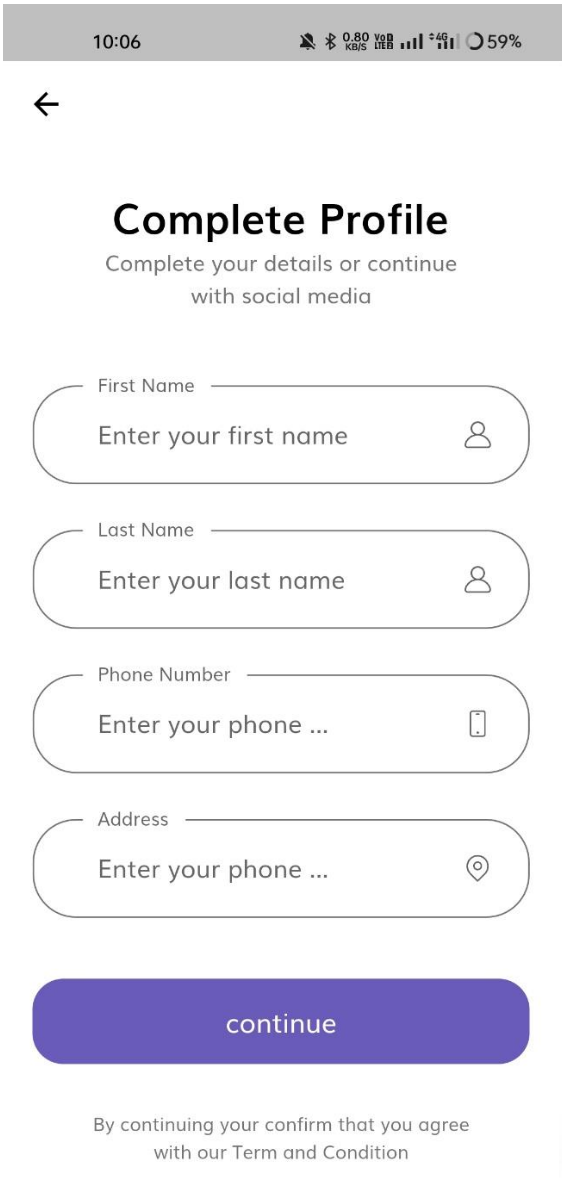


FIGURE-5.1.4 Home Screen



Complete Profile

Complete your details or continue with social media

Enter your first name

Enter your last name

Enter your phone ...

Enter your phone ...

By continuing your confirm that you agree with our Term and Condition

FIGURE-5.1.5 Complete details Screen

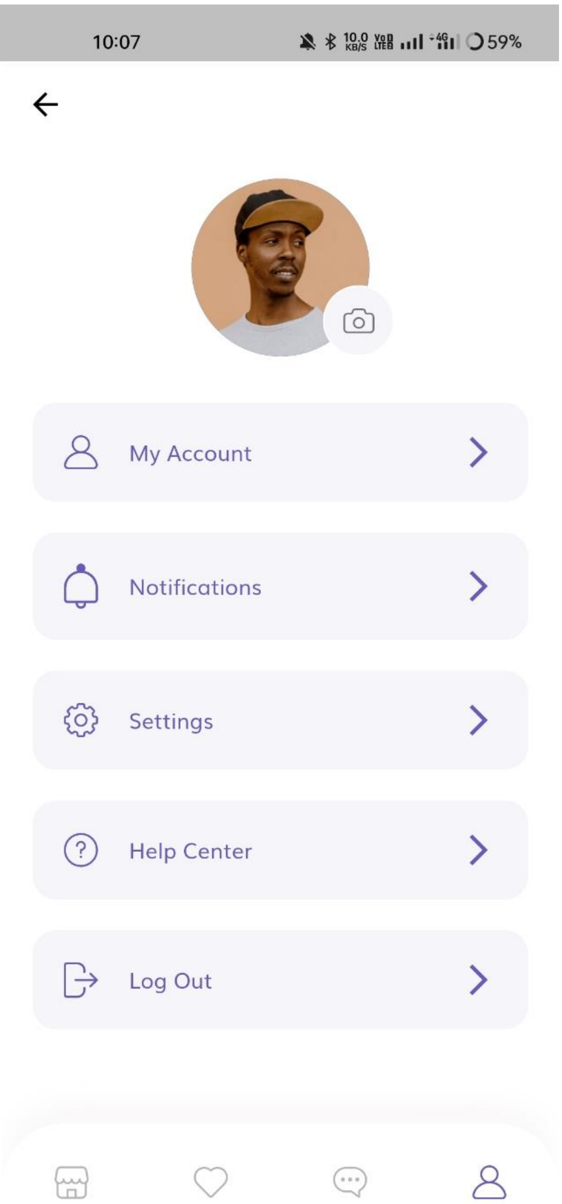
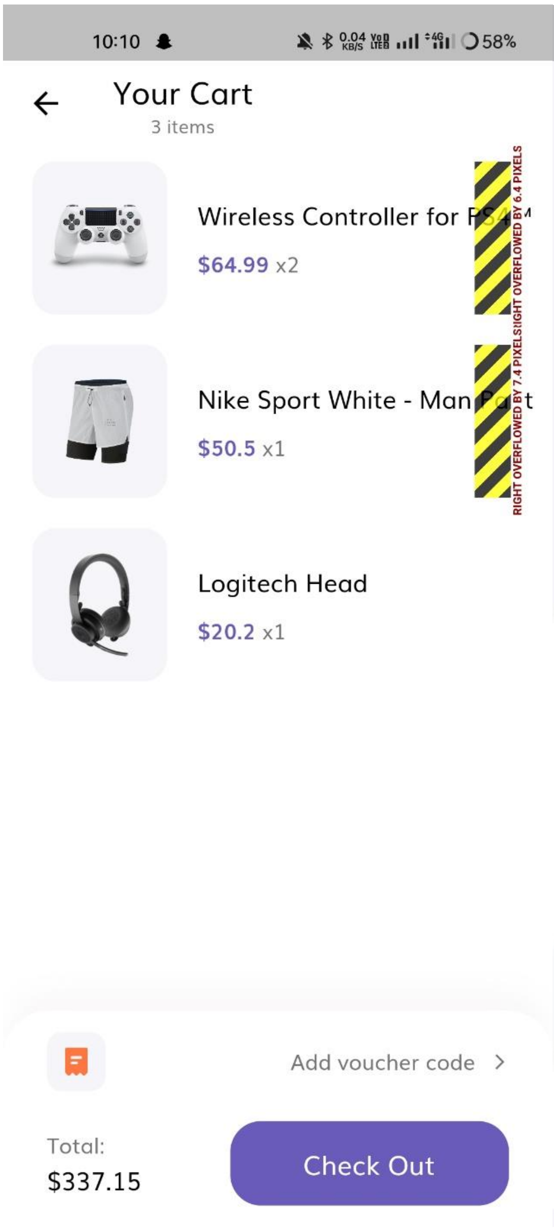


FIGURE-5.1.6 My profile Screen



Your Cart

3 items

Wireless Controller for PS4™
$64.99 x2

Nike Sport White - Man Pant
$50.5 x1

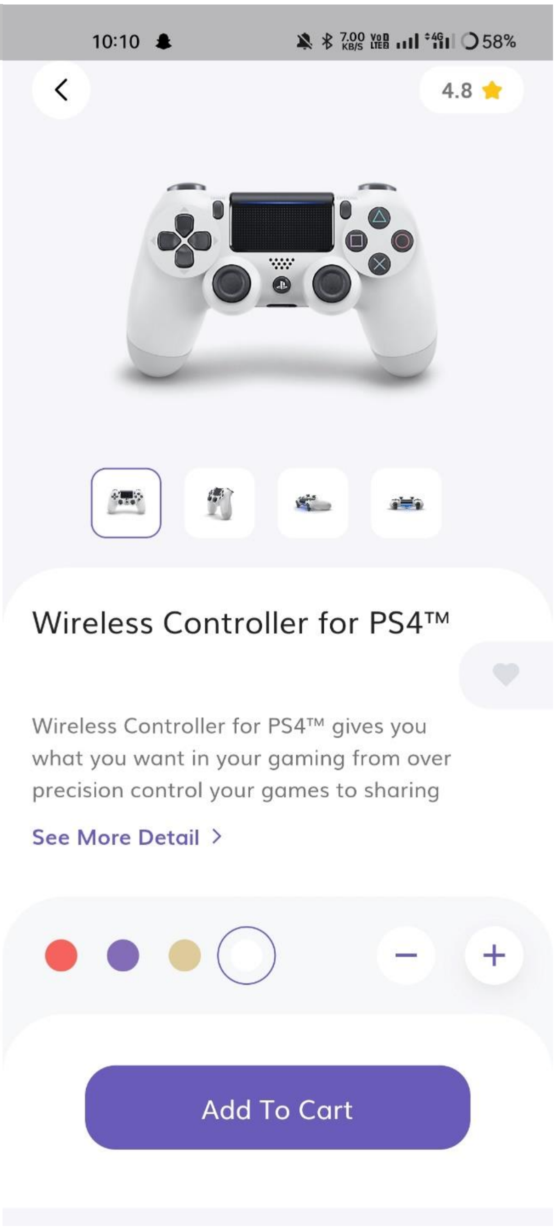
Logitech Head
$20.2 x1

RIGHT OVERFLOWED BY 6.4 PIXELS
RIGHT OVERFLOWED BY 7.4 PIXELS

Add voucher code >

Total:
$337.15

Check Out



4.8 ★

Wireless Controller for PS4™

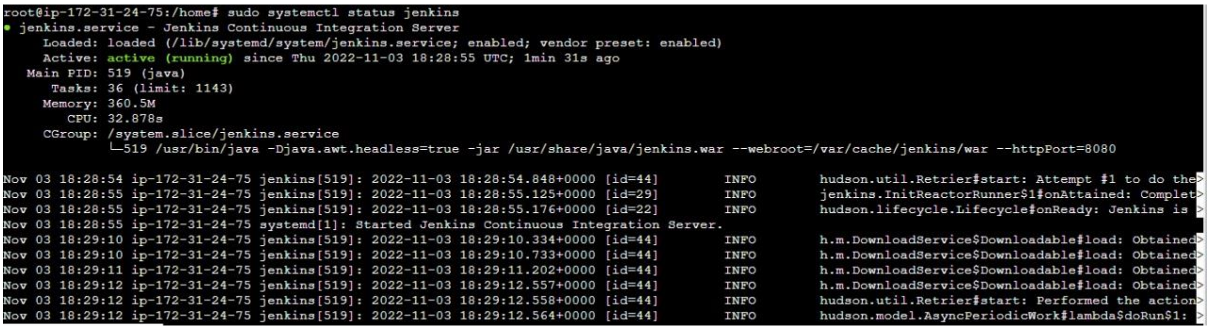
Wireless Controller for PS4™ gives you what you want in your gaming from over precision control your games to sharing

See More Detail >

Add To Cart

FIGURE-5.1.7 Add to Cart Screen

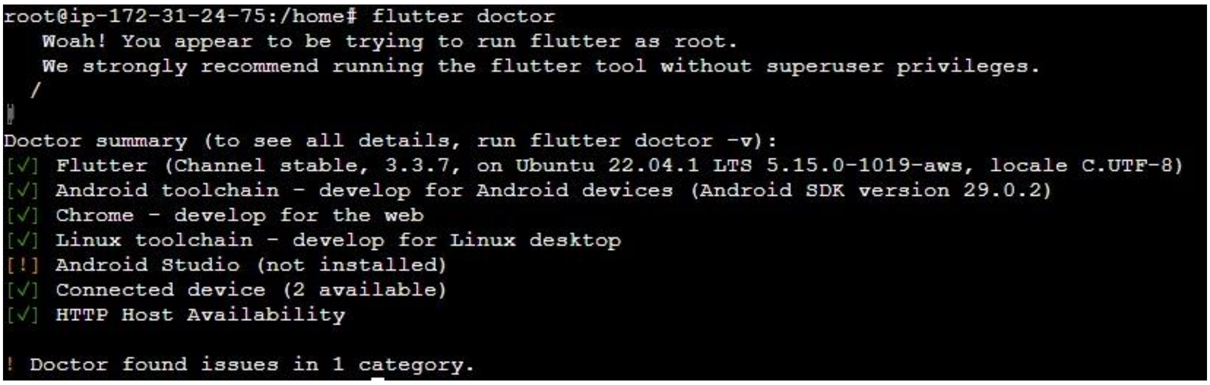
FIGURE-5.1.8 Product details Screen



```
root@ip-172-31-24-75:/home# sudo systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
     Loaded: loaded (/lib/systemd/system/jenkins.service; enabled; vendor preset: enabled)
     Active: active (running) since Thu 2022-11-03 18:28:55 UTC; 1min 31s ago
   Main PID: 519 (java)
      Tasks: 36 (limit: 1143)
     Memory: 360.5M
        CPU: 32.878s
     CGroup: /system.slice/jenkins.service
             └─519 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenkins/war --httpPort=8080

Nov 03 18:28:54 ip-172-31-24-75 jenkins[519]: 2022-11-03 18:28:54.848+0000 [id=44]        INFO       hudson.util.Retrier#start: Attempt #1 to do the>
Nov 03 18:28:55 ip-172-31-24-75 jenkins[519]: 2022-11-03 18:28:55.125+0000 [id=29]        INFO       jenkins.InitReactorRunner$1#onAttained: Complet>
Nov 03 18:28:55 ip-172-31-24-75 jenkins[519]: 2022-11-03 18:28:55.176+0000 [id=22]        INFO       hudson.lifecycle.Lifecycle#onReady: Jenkins is >
Nov 03 18:28:55 ip-172-31-24-75 systemd[1]: Started Jenkins Continuous Integration Server.
Nov 03 18:29:10 ip-172-31-24-75 jenkins[519]: 2022-11-03 18:29:10.334+0000 [id=44]        INFO       h.m.DownloadService$Downloadable#load: Obtained>
Nov 03 18:29:10 ip-172-31-24-75 jenkins[519]: 2022-11-03 18:29:10.733+0000 [id=44]        INFO       h.m.DownloadService$Downloadable#load: Obtained>
Nov 03 18:29:11 ip-172-31-24-75 jenkins[519]: 2022-11-03 18:29:11.202+0000 [id=44]        INFO       h.m.DownloadService$Downloadable#load: Obtained>
Nov 03 18:29:12 ip-172-31-24-75 jenkins[519]: 2022-11-03 18:29:12.557+0000 [id=44]        INFO       h.m.DownloadService$Downloadable#load: Obtained>
Nov 03 18:29:12 ip-172-31-24-75 jenkins[519]: 2022-11-03 18:29:12.558+0000 [id=44]        INFO       hudson.util.Retrier#start: Performed the action>
Nov 03 18:29:12 ip-172-31-24-75 jenkins[519]: 2022-11-03 18:29:12.564+0000 [id=44]        INFO       hudson.model.AsyncPeriodicWork#lambda$doRun$1: >
```

FIGURE-5.1.9 Jenkins Setup on Server

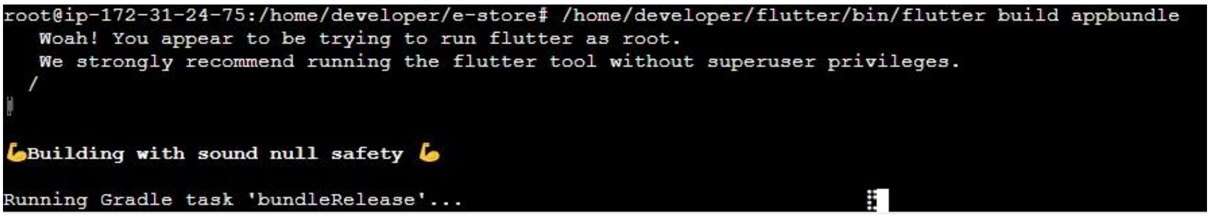


```
root@ip-172-31-24-75:/home# flutter doctor
   Woah! You appear to be trying to run flutter as root.
   We strongly recommend running the flutter tool without superuser privileges.
  /

Doctor summary (to see all details, run flutter doctor -v):
[√] Flutter (Channel stable, 3.3.7, on Ubuntu 22.04.1 LTS 5.15.0-1019-aws, locale C.UTF-8)
[√] Android toolchain - develop for Android devices (Android SDK version 29.0.2)
[√] Chrome - develop for the web
[√] Linux toolchain - develop for Linux desktop
[!] Android Studio (not installed)
[√] Connected device (2 available)
[√] HTTP Host Availability

! Doctor found issues in 1 category.
```

FIGURE-5.1.10 Flutter Setup on Server



```
root@ip-172-31-24-75:/home/developer/e-store# /home/developer/flutter/bin/flutter build appbundle
   Woah! You appear to be trying to run flutter as root.
   We strongly recommend running the flutter tool without superuser privileges.
  /

💪Building with sound null safety 💪

Running Gradle task 'bundleRelease'...
```

FIGURE-5.1.11 Build in Running State

5.2 Limitations

- New skill must be learned
- Steep learning curve to implement automation
- Big upfront investment
- Legacy systems rarely support CI/CD
- High degree of discipline and dedication to quality.

CHAPTER-6 PROJECT OUTCOME AND FUTURE ENHANCEMENTS

6.1 Project Outcome

Continuous integration and continuous delivery provide an ideal scenario for your organization’s application teams. Your developers simply push code to a repository. This code will be integrated, tested, deployed, tested again, merged with infrastructure, go through security and quality reviews, and be ready to deploy with extremely high confidence.

When CI/CD is used, code quality is improved and software updates are delivered quickly and with high confidence that there will be no breaking changes. The impact of any release can be correlated with data from production and operations. It can be used for planning the next cycle, too—a vital DevOps practice in your organization’s cloud transformation.

6.2 Future Enhancements

Today, Jenkins is the most popular tool to implement continuous integration in DevOps projects. There are various reasons why it has become so hugely popular.

Now, the industry needs to streamline testing by reducing development tool complexity. Tools are generating ample data, but the next evolution is efficiently taking that data and using it to drive DevOps to even faster CI/CD cycles.

CI/CD continuously merges codes and continuously deploys them to production after thorough testing, keeping the code in a release-ready state. It's important to have as part of deployment a production environment set up that closely mimics that which end-users will ultimately be using.

CHAPTER-7 CONCLUSION

6.1 Conclusion

UI is simple and minimalistic, understandable for everyone. It greatly saves the time. We can give more advance software for E-Commerce Store including more facilities. Finally, the system is implemented and tested according to test cases. We will attempt to advance this application as it could contact an ever-increasing number of individuals.

E-Store is a devops project which assist the user to deploy their application, with the help of devops technology which include the Jenkins tool and therefore it will push it into the main server.

References

1. https://www.youtube.com/watch?v=eX3ZimWWHh4&list=PLBGx66SQNZ8aPsFDwb79JrS2KQBTIZo10
2. https://www.digitalocean.com/community/tutorials/how-to-install-jenkins-on-ubuntu-20-04
3. https://ubuntu.com/blog/getting-started-with-flutter-on-ubuntu
4. https://us-west-2.console.aws.amazon.com/ec2/home?region=us-west-2#Instances
5. https://docs.flutter.dev/deployment/cd