

# **Rush App iOS Automation**

Major project report submitted in partial fulfillment of the requirement  
for the degree of Bachelor of Technology

in

**Computer Science and Engineering**

By

**Harshit Kumar Gupta (181334)**

UNDER THE SUPERVISION OF

**Mr. Praveen Modi**



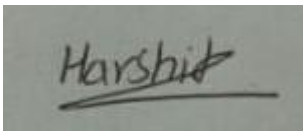
Department of Computer Science & Engineering and  
Information Technology

**Jaypee University of Information Technology, Wagnaghat,  
173234, Himachal Pradesh, IND**

## Candidate's Declaration

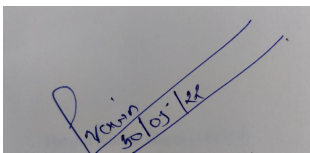
I hereby declare that the work presented in this report entitled "**Hike Internship Report**" in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Engineering/Information Technology** submitted in the department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology Waknaghat is an authentic record of my own work carried out over a period from February 2022 to June 2022 under the supervision of **Praveen Modi** (Assistant Professor, Computer Science Engineering).

The matter embodied in the report has not been submitted for the award of any other degree or diploma.



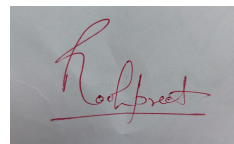
Harshit Kumar Gupta, 181334

This is to certify that the above statement made by the candidate is true to the best of my knowledge.



(College Supervisor Signature)

Praveen Modi  
Assistant Professor  
Computer Science Engineering  
Dated: 25-05-2022



Roohpreet Kaur  
SDET  
Hike  
Dated: 25-05-2022

## **ACKNOWLEDGEMENT**

First, I express my deepest gratitude and gratitude to Almighty God for His divine blessing enabling us to complete the project successfully.

Thank you very much and wish my deepest thanks to Supervisor Mr Praveen Modi, Assistant Professor (SG), Department of CSE Jaypee University of Information Technology, Waknaghat. His enduring patience, expert guidance, constant encouragement, constant watchfulness and enthusiasm, constructive criticism, valuable advice, reading a lot of low-level writing and editing at all stages made it possible to complete the project.

I would like to extend my deepest gratitude to Mr Praveen Modi, Department of CSE, for his kind assistance in completing my work.

I would also kindly welcome each of those people who have helped me directly or indirectly in making this project a success. In this unique situation, I would like to thank the various staff, educators and non-instructors, who have helped me in my work.

Finally, I must admit that I have a lot of respect for the support of my parents, teachers and fellow classmates.

Harshit Kumar Gupta (181334)

## **ABSTRACT**

We believe that the rush of the game matters more than the outcome. A rush that comes from competing on the strength of our skills, where every win is deserved, and any loss is simply training ground to play the next game better. We believe that we aren't just a platform for users to win money in games, but a community of players with integrity, who are committed to sportsmanship, transparency, and fair play.

## **TABLE OF CONTENTS**

<b>S.No</b>	<b>Topic</b>	<b>Page No.</b>
1.	Introduction	7
2.	Problem Statement	8
3.	Scope and Objective	9
4.	Existing Data	10-11
5.	Software Requirements Analysis	12-19
6.	Software Design	23-24
7.	Coding Snippets	25-26
8.	Project Structure	27
9.	Testing Techniques	20-22
10.	Project Outcome	28
11.	Conclusion	29
12.	References	30

## **Introduction**

As Hike, a trusted Indian youth-centric brand, we have years of experience in building innovative, high-quality technology products for mobile users in emerging markets. We ensure that every aspect of the product is of the highest quality, rigorously tested, and designed to work exactly as expected. At the heart of Rush is our community of players, where every player has a HikeMojiled identity that not just ensures safety and privacy, but also adds to the fun of gameplay through innovative self-expression. We respect the trust that our community places in us and go the extra mile to ensure that we are transparent in all the experiences we offer - managing the player's online wallet securely, matching players with other skilled players (no bots!), and our 24-hour customer service to address any specific requests.

## **Problem Statement**

Design an online gaming platform where the competitive spirit of India can come alive. People can play against each other in skill-games and win real money

## **Scope and Objective**

The main aim is to have all our games as skill based. It is the player's mettle that should determine wins. To build a community of real players only. To have no bots. The HikeMojji avatars add excitement and expression to the game where every emotion can be shared uniquely - creating a sense of community with fellow players. For the first time in India, to introduce a VIP membership plan where members can save up to 20% on the entry fee in each game.



## Existing Literature

First and foremost, in skill gaming, the most important factor that any gamer would want to master would be the skill, strategies and in-depth knowledge of a particular game he/she is playing — as at the end of the day, to win or to perform optimally, skill and competence are what obviously will matter the most.

However, in any type of online gambling, the predominant factor to win money is fate or probability (counting on the odds) — which means even if you are skilled, it does not guarantee you success in gambling. Secondly, the motivation of gamers no matter whether he/she is an amateur or a pro — is to play a game or enter a gaming tournament for the fun, relaxation and/or social element of it, rather than only for earning money. The gaming ecosystem thrives on immense passion, engagement, and vibrancy.

On the other hand, in gambling sites or apps, one would usually go only for satiating the desire (or greed) to get-rich-quick and earn more and more money, and furthermore he/she would also be required (in most cases) to deposit a certain amount before being allowed to start gambling.

In India, monetary stakes (wagers) are usually placed on online card games like Rummy, Poker, Teen Patti and Fantasy games like Fantasy Cricket and Fantasy Soccer — although this does not necessarily imply that playing or taking part in these ‘real money’-based games makes one a gambler!

We need to understand the distinction between ‘games of chance’ and ‘games of skill’ — given the current scenario in India wherein in most States, betting money on a game of chance is deemed illegal, but placing wagers on a game of skill is, in fact, legal in many places (or at least can be legally defended in a court of law).

A 'game of skill' is one where a player's success is determined on the basis of how efficiently he/she plays the game and how much he is able to put his/her skills to use. Whereas in most 'games of chance', the gameplay and even winners are randomly determined, either manually or via technology (maybe through an algorithm or a specified programming), and therefore the winning and the prize has majorly got to do with the luck of a player.

In 2019, in the Ravindra Singh Chaudhary vs. Union of India case, the Rajasthan High Court had ruled that Fantasy Sports did not constitute gambling as its players would need skills similar to that of a real sports team manager. In a recent ruling, the Tamil Nadu High Court has struck down the State's ban on online card games like Rummy, Poker etc. These are welcome developments related to differentiating 'games of chance' and 'games of skill', and shall go a long way in promoting a positive narrative in favor of gaming against gambling.

## Software Requirements Analysis

1. Language:

Java:



java is used in server side automation.

- a) Java version "1.8.0\_251"
- b) Java(TM) SE Runtime Environment (build 1.8.0\_251-b08)
- c) Java HotSpot(TM) 64-Bit Server VM (build 25.251-b08, mixed mode)

Swift:



# Swift

User in iOS automation. The Framework Used Is XCUITest.

Swift is a new programming language for iOS, macOS, watchOS, and tvOS app development. Nonetheless, many parts of Swift will be familiar from your experience of developing in C and Objective-C.

## 2. IDE: IntelliJ



# IntelliJ IDEA

- a) Clone the repository in a working directory. This step will require the secret key that was obtained while setting up github.
- b) Clone directly from File Structure
- c) Clone through VCS. Prefer VCS to avoid confusions.

## 3. Figma:



Figma is a browser-based UI and UX design application. Whenever a new feature comes, you can see it's design on figma.

#### 4. Jenkins:



# Jenkins

An open source automation server which enables developers to reliably build, test, and deploy their software. When it comes to test automation, Jenkins provides plugins that help run test suites, gather and dashboard results, and provide details on failures.

#### 5. CircleCI:



a) CircleCI is a continuous integration & delivery platform that helps the development teams to release code rapidly and automate the build, test, and deploy.

b) After repositories on Github are authorized and added as a project to circleci.com, every code triggers CircleCI runs jobs. CircleCI also sends an email notification of success or failure after the tests/checks complete.

#### 6. Clover, Coverage and Merge Reports:

a) Sonar is a Code Quality Assurance tool that collects and analyzes source code, and provides reports for the code quality of the project. Also highlights the complex areas of code that are less covered by tests.

b) Code coverage is a metric that we use to check the quality of our tests, as it represents the percentage of production code that has been tested.

## 7. Jira:



JIRA is used for issue tracking, bug tracking and project management.

### **To create a Task in JIRA**

- Click on Create at the top of the screen
- Select Tools&Automation Project
- Select Task as issue type
- Write a Summary for the task and click on Assign to me
- Select platform as iOS and click on Create button

### **To raise a Bug**

- Click on Create at the top of the screen
- Select Vibe/Rush Project
- Select Bug as issue type
- Write a Summary for the task and click on the pencil icon to edit Default Assignee.
- Select Priority, fix version, and choose *Automation* in Labels
- Select platform as iOS
- Fill in the description as precisely as possible. Attach screenshots if needed

## 8. Crashlytics:



Firebase Crashlytics is a realtime crash reporter that helps you track, prioritise, and fix crashes to improve the app quality. It does this through three aspects:

### a) Logs:

Log events in your app to be sent with the crash report for context if your app crashes.

### b) Crash reports:

Every crash is automatically turned into a crash report and sent when the application next opens.

### c) Stack traces:

Even when an error is caught and your app recovers, the Dart stack trace can still be sent.

## 9. Databases:

### a) MongoDB:



MongoDB is a cross-platform, document oriented database that provides, high performance, high availability, and easy scalability. MongoDB works on concept of collection and document.

### i. Database:

Database is a physical container for collections. Each database gets its own set of files on the file system. A single MongoDB server typically has multiple databases.

ii. Collection: Collection is a group of MongoDB documents. It is the equivalent of an RDBMS table. A collection exists within a single database. Collections do not enforce a schema. Documents within a collection can have different fields. Typically, all documents in a collection are of similar or related purpose.

iii. Document: A document is a set of key-value pairs. Documents have dynamic schema. Dynamic schema means that documents in the same collection do not need to have the same set of fields or structure, and common fields in a collection's documents may hold different types of data.

b) MySQL:



MySQL is a Relational Database Management System (RDBMS) software that provides many things, which are as follows:

i. It allows us to implement database operations on tables, rows, columns, and indexes.

ii. It defines the database relationship in the form of tables (collection of rows and columns), also known as relations.

iii. It provides the Referential Integrity between rows or columns of various tables.

iv. It allows us to update the table indexes automatically.

v. It uses many SQL queries and combines useful information from multiple tables for the end-users.



## 10. Redis:



Redis is an open source (BSD licensed), in-memory data structure store, used as a database, cache, and message broker. Redis provides data structures such as strings, hashes, lists, sets, sorted sets with range queries, bitmaps, hyperloglogs, geospatial indexes, and streams. Redis has built-in replication, Lua scripting, LRU eviction, transactions, and different levels of on-disk persistence, and provides high availability via Redis Sentinel and automatic partitioning with Redis Cluster.

## 11. Zookeeper:

ZooKeeper is an open source Apache project that provides a centralized service for providing configuration information, naming, synchronization and group services over large clusters in distributed systems. The goal is to make these systems easier to manage with improved, more reliable propagation of changes.

## 12. Postman:



a) Postman is a standalone software testing API (Application Programming Interface) platform to build, test, design, modify, and document APIs. It is a simple Graphic User Interface for sending and viewing HTTP requests and responses.

b) While using Postman, for testing purposes, one doesn't need to write any HTTP client network code. Instead, we build test suites called collections and let Postman interact with the API.

c) In this tool, nearly any functionality that any developer may need is

embedded. This tool has the ability to make various types of HTTP requests like GET, POST, PUT, PATCH, and convert the API to code for languages like JavaScript and Python.

### 13. Charles Proxy:



Charles Web Debugging Proxy is a cross-platform HTTP debugging proxy server application written in Java. It enables the user to view HTTP, HTTPS, HTTP/2[3] and enabled TCP port traffic accessed from, to, or via the local computer. This includes requests and responses including HTTP headers and metadata (e.g. cookies, caching and encoding information) with functionality targeted at assisting developers analyze connections and messaging.

### 15. JUnit:

JUnit is a unit testing framework for Java programming language. It plays a crucial role test-driven development, and is a family of unit testing frameworks collectively known as xUnit. JUnit promotes the idea of "first testing then coding", which emphasizes on setting up the test data for a piece of code that can be tested first and then implemented. This approach is like "test a little, code a little, test a little, code a little." It increases the productivity of the programmer and the stability of program code, which in turn reduces the stress on the programmer and the time spent on debugging.

### 16. TestNg:

TestNG is an automation testing framework in which NG stands for "Next Generation". TestNG is inspired by JUnit which uses the annotations (@). TestNG overcomes the disadvantages of JUnit and is designed to make end-to-end testing easy. Using TestNG, you can generate a proper report, and you can easily come to know how many

test cases are passed, failed, and skipped. You can execute the failed test cases separately.

#### 17. Spring Boot:

Java Spring Framework (Spring Framework) is a popular, open source, enterprise-level framework for creating standalone, production-grade applications that run on the Java Virtual Machine (JVM). 17 Java Spring Boot (Spring Boot) is a tool that makes developing web application and microservices with Spring Framework faster and easier through three core capabilities:

- a) Autoconfiguration
  - b) An opinionated approach to configuration
  - c) The ability to create standalone applications
- These features work together to provide you with a tool that allows you to set up a Spring-based application with minimal configuration and setup.

#### 18. XCUITest:

When there are so many mobile test automation frameworks on the market, you might be asking why you should study XCUITest. It's because XCUITest is native, which means it's lightning fast. XCUITest is also used by all of the mobile test frameworks.

If you're using XCUITest to automate your iOS or other Apple platform apps, you're probably writing your tests in Swift, which is a very comparable stack to what your developers are using, making it easier for app developers and automation engineers to collaborate on XCUITest.

However, there are a couple drawbacks to using XCUITest. You can't develop tests for various platforms using XCUITest. You can't, for example, have a test that compares iOS and Android apps.

#### 19. Git:

GIT is a version control system. It is a centralised repository to keep your code safe and handy. Benefit of GIT is that every member in your organisation can

fetch your code and use it.

### Git Commands: Working With Local Repositories

- `git init`
- The command `git init` is used to create an empty Git repository.
- After the `git init` command is used, a `.git` folder is created in the directory with some subdirectories. Once the repository is initialized, the process of creating other files begins.

`git init`

```
SL-LP-DNS-0223+Taha@SL-LP-DNS-0223 MINGW64 ~/Git_demo/FirstRepo
$ git init
Initialized empty Git repository in C:/Users/Taha/Git_demo/FirstRepo/.git/
SL-LP-DNS-0223+Taha@SL-LP-DNS-0223 MINGW64 ~/Git_demo/FirstRepo (master)
$ |
```

- `git add`
- Add command is used after checking the status of the files, to add those files to the staging area.
- Before running the commit command, "`git add`" is used to add any new or modified files.


`git add .`

```
Untracked files:
(use "git add <file>..." to include in what will be committed)
alpha.txt

nothing added to commit but untracked files present (use "git add" to track)
SL-LP-DNS-0223+Taha@SL-LP-DNS-0223 MINGW64 ~/Git_demo/FirstRepo (master)
$ git add .
SL-LP-DNS-0223+Taha@SL-LP-DNS-0223 MINGW64 ~/Git_demo/FirstRepo (master)
$
```

- git commit
- The commit command makes sure that the changes are saved to the local repository.
- The command "git commit -m <message>" allows you to describe everyone and help them understand what has happened.

```
git commit -m "commit message"
```



```
SL-LP-DNS-0223+Taha@SL-LP-DNS-0223 MINGW64 ~/Git_demo/FirstRepo (master)
$ git status
On branch master

No commits yet

changes to be committed:
  (use "git rm --cached <file>..." to unstage)
   new file:   alpha.txt

SL-LP-DNS-0223+Taha@SL-LP-DNS-0223 MINGW64 ~/Git_demo/FirstRepo (master)
$ git commit -m "alpha"
[master (root-commit) b89b00a] alpha
 1 file changed, 1 insertion(+)
 create mode 100644 alpha.txt

SL-LP-DNS-0223+Taha@SL-LP-DNS-0223 MINGW64 ~/Git_demo/FirstRepo (master)
$
```

- git status
- The git status command tells the current state of the repository.
- The command provides the current working branch. If the files are in the staging area, but not committed, it will be shown by the git status. Also, if there are no changes, it will show the message no changes to commit, working directory clean.

```
git status
```

```
SL-LP-DNS-0223+Taha@SL-LP-DNS-0223 MINGW64 ~/Git_demo/FirstRepo (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        alpha.txt

nothing added to commit but untracked files present (use "git add" to track)
SL-LP-DNS-0223+Taha@SL-LP-DNS-0223 MINGW64 ~/Git_demo/FirstRepo (master)
$
```

```
SL-LP-DNS-0223+Taha@SL-LP-DNS-0223 MINGW64 ~/Git_demo/FirstRepo (master)
$ git status
On branch master
nothing to commit, working tree clean
SL-LP-DNS-0223+Taha@SL-LP-DNS-0223 MINGW64 ~/Git_demo/FirstRepo (master)
$
```

- git config
- The git config command is used initially to configure the user.name and user.email. This specifies what email id and username will be used from a local repository.
- When git config is used with --global flag, it writes the settings to all repositories on the computer.

git config --global user.name "any user name"

git config --global user.email <email id>

```
SL-LP-DNS-0223+Taha@SL-LP-DNS-0223 MINGW64 ~
$ git config --global user.name "simplilearn github"

SL-LP-DNS-0223+Taha@SL-LP-DNS-0223 MINGW64 ~
$ git config --global user.email "simplilearn github"

SL-LP-DNS-0223+Taha@SL-LP-DNS-0223 MINGW64 ~
$ git config --list
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=openssl
http.sslcainfo=C:/Users/Taha/AppData/Local/Programs/Git/mingw64/ssl/certs/ca-bundle.crt
core.autocrlf=true
core.fsmonitor=true
core.symlinks=false
credential.helper=manager
user.name=Simplilearn Github
user.email=Simplilearn github
user.name=simplilearn github

SL-LP-DNS-0223+Taha@SL-LP-DNS-0223 MINGW64 ~
$
```

- git branch
- The git branch command is used to determine what branch the local repository is on.
- The command enables adding and deleting a branch.

# Create a new branch

```
git branch <branch_name>
```

# List all remote or local branches

```
git branch -a
```

# Delete a branch

```
git branch -d <branch_name>
```

- 

git checkout

- The git checkout command is used to switch branches, whenever the work is to be started on a different branch.
- The command works on three separate entities: files, commits, and branches.

# Checkout an existing branch

```
git checkout <branch_name>
```

# Checkout and create a new branch with that name

```
git checkout -b <new_branch>
```

- git merge
- The git merge command is used to integrate the branches together. The command combines the changes from one branch to another branch.
- It is used to merge the changes in the staging branch to the stable branch.



## Testing Techniques

Automation Testing or Test Automation is a software testing technique that performs using special automated testing software tools to execute a test case suite. On the contrary, Manual Testing is performed by a human sitting in front of a computer carefully executing the test steps. The automation testing software can also enter test data into the System Under Test, compare expected and actual results and generate detailed test reports. Software Test Automation demands considerable investments of money and resources.

Successive development cycles will require execution of same test suite repeatedly. Using a test automation tool, it's possible to record this test suite and re-play it as required. Once the test suite is automated, no human intervention is required. This improved ROI of Test Automation. The goal of Automation is to reduce the number of test cases to be run manually and not to eliminate Manual Testing altogether.

Test Automation is the best way to increase the effectiveness, test coverage, and execution speed in software testing. Automated software testing is important due to the following reasons:

- Manual Testing of all workflows, all fields, all negative scenarios is time and money consuming
- It is difficult to test for multilingual sites manually
- Test Automation in software testing does not require Human intervention. You can run automated test unattended (overnight)
- Test Automation increases the speed of test execution
- Automation helps increase Test Coverage • Manual Testing can become boring and hence error-prone.

Test cases to be automated can be selected using the following criterion to increase the automation ROI

- High Risk – Business Critical test cases
- Test cases that are repeatedly executed
- Test Cases that are very tedious or difficult to perform manually
- Test Cases which are time-consuming The following category of test cases are not suitable for automation:
- Test Cases that are newly designed and not executed manually at

least once

- Test Cases for which the requirements are frequently changing
- Test cases which are executed on an ad-hoc basis.

## **Regression Testing**

Regression Testing is defined as a type of software testing to confirm that a recent program or code change has not adversely affected existing features. Regression Testing is nothing but a full or partial selection of already executed test cases which are re-executed to ensure existing functionalities work fine.

This testing is done to make sure that new code changes should not have side effects on the existing functionalities. It ensures that the old code still works once the latest code changes are done.

The Need of Regression Testing mainly arises whenever there is requirement to change the code and we need to test whether the modified code affects the other part of software application or not. Moreover, regression testing is needed, when a new feature is added to the software application and for defect fixing as well as performance issue fixing.

## **Integration Testing**

Integration Testing is defined as a type of testing where software modules are integrated logically and tested as a group. A typical software project consists of multiple software modules, coded by different programmers. The purpose of this level of testing is to expose defects in the interaction between these software modules when they are integrated.

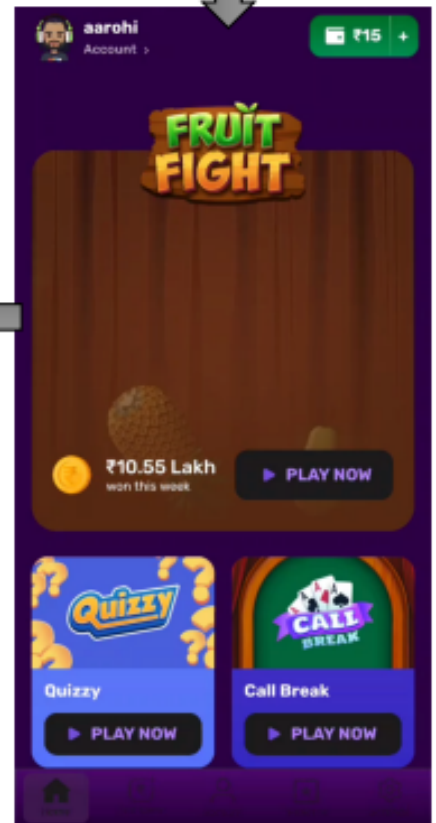
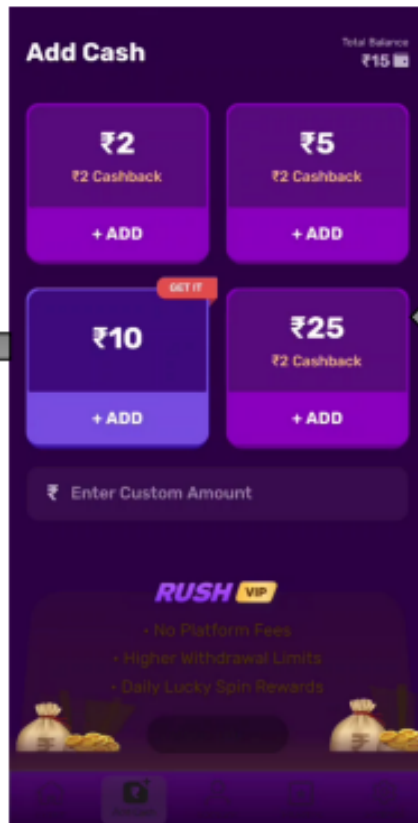
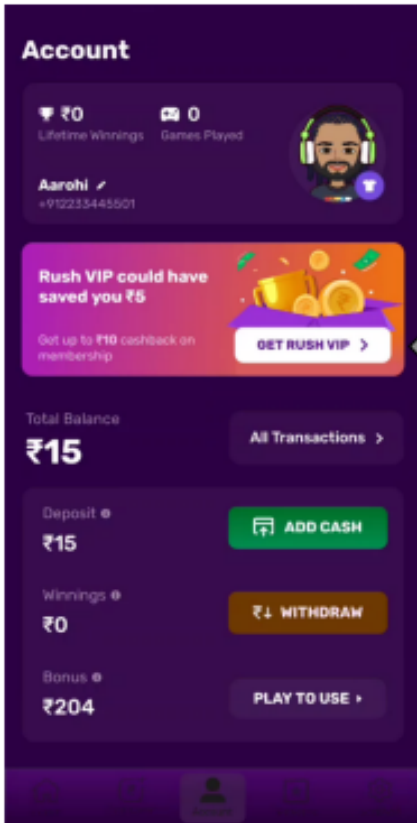
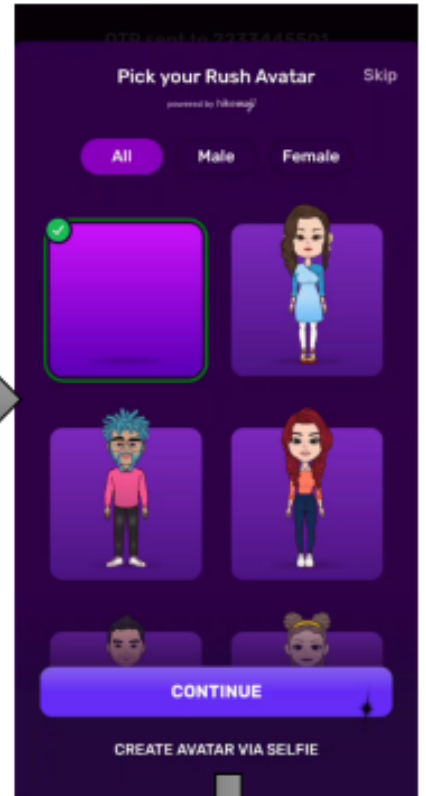
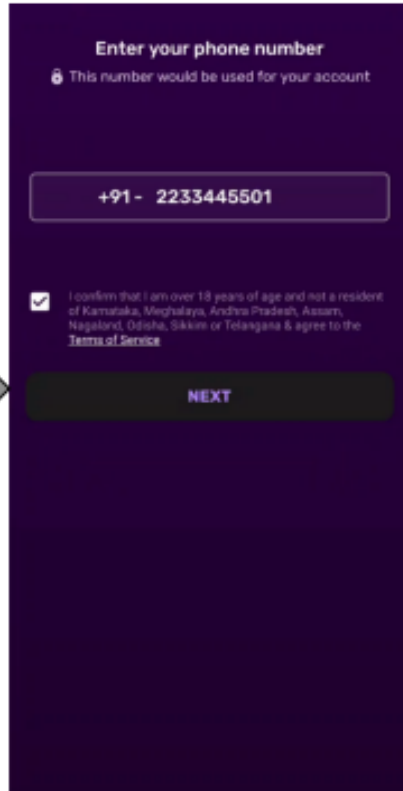
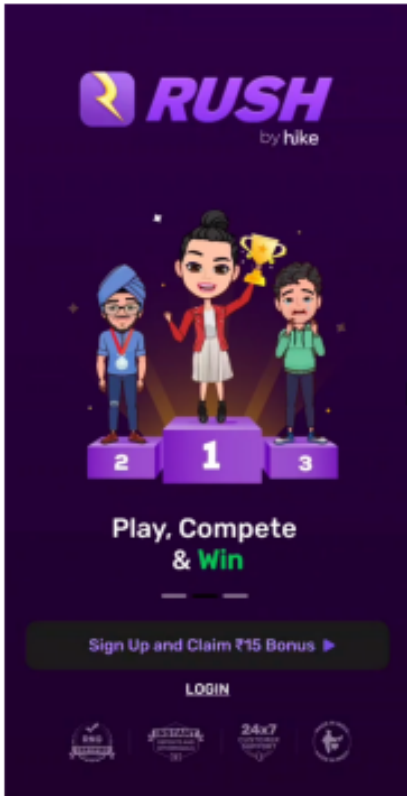
Integration Testing focuses on checking data communication amongst these modules. Hence it is also termed as 'I & T' (Integration and Testing), 'String Testing' and sometimes 'Thread Testing'.

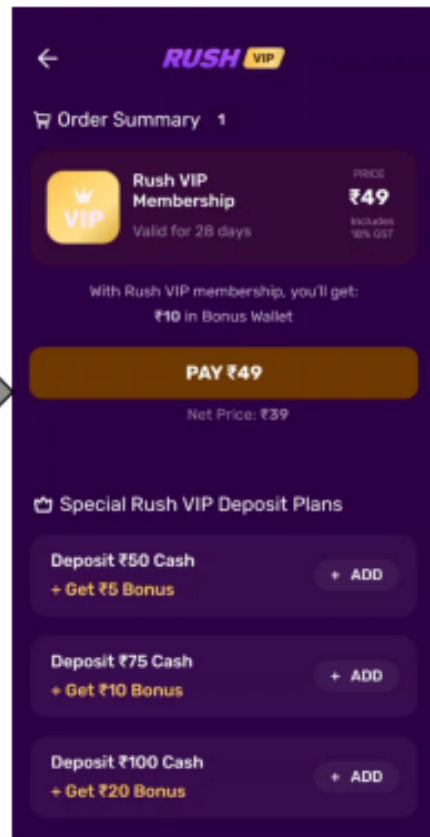
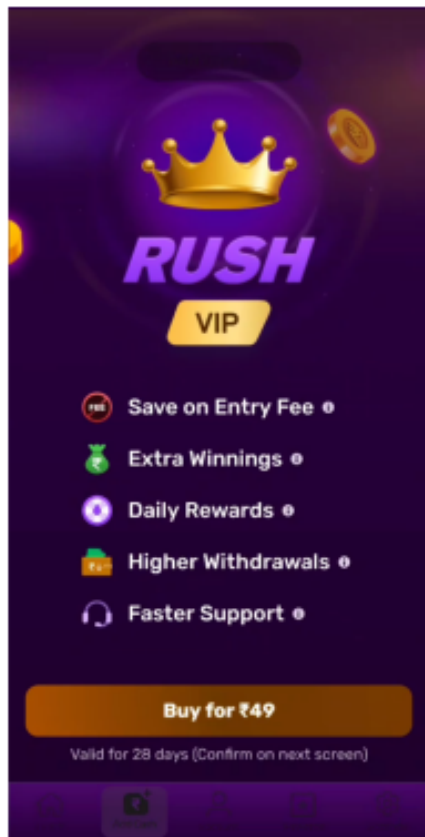
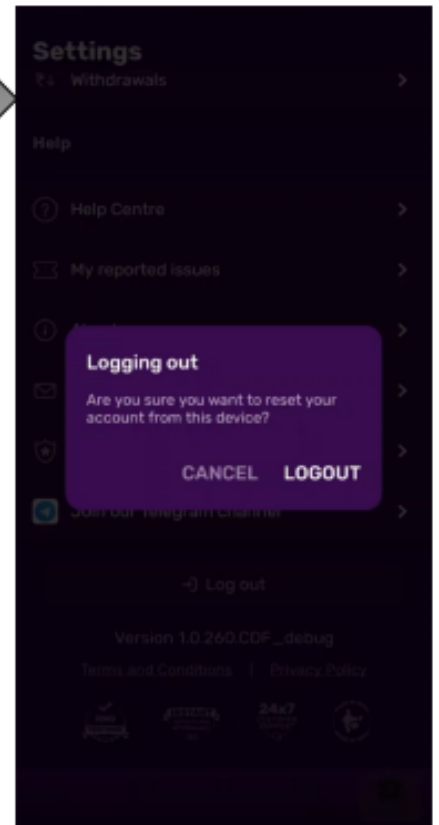
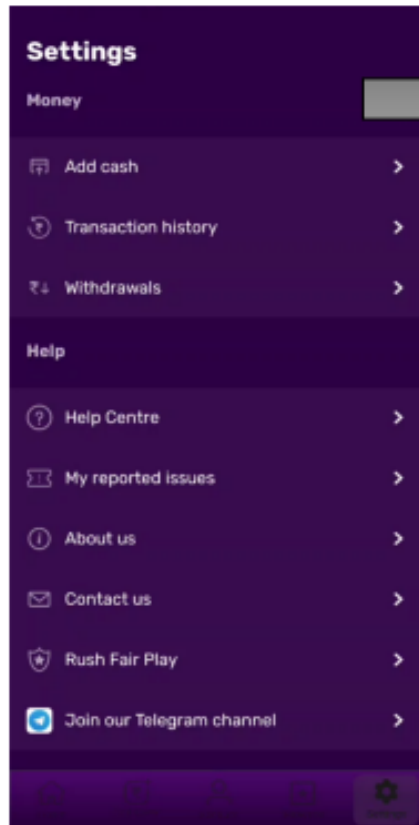
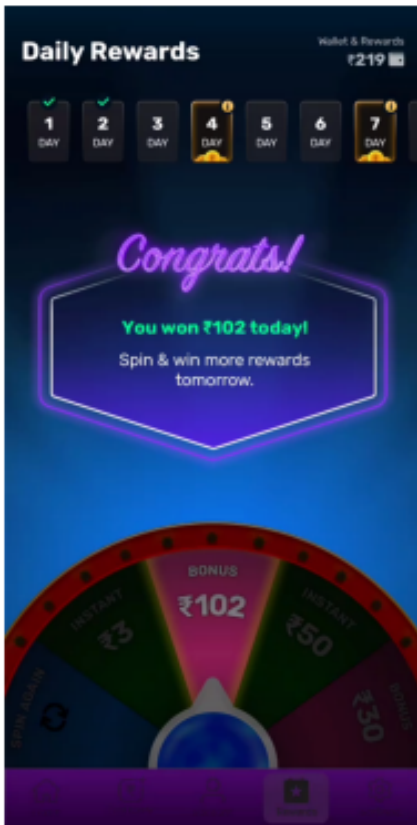
## **Unit Testing**

Unit Testing is a type of software testing where individual units or components of a software are tested. The purpose is to validate that each unit of the software code performs as expected. Unit Testing is done during the development (coding phase) of an application by the developers. Unit Tests isolate a section of code and verify its correctness. A unit may be an individual function, method, procedure, module, or object.

In SDLC, STLC, V Model, Unit testing is the first level of testing done before integration testing. Unit testing is a WhiteBox testing technique that is usually performed by the developer. Though, in a practical world due to time crunch or reluctance of developers to tests, QA engineers also do unit testing.

Unit Testing is important because software developers sometimes try saving time doing minimal unit testing and this is myth because inappropriate unit testing leads to high cost Defect fixing during System Testing, Integration Testing and even Beta Testing after application is built. If proper unit testing is done in early development, then it saves time and money in the end.





# Coding Snippets

## Individual Test Cases

```
@Test(groups = {"infra", "raf"})
public void verifyChildSignUpForFreeParent() throws Exception {
    deleteUsers = false;
    UserInfo child = RushReferralService.getInstance().createEligibleUserForRaf().getFirst();
    UserInfo parent = RushReferralService.getInstance().createEligibleUserForRaf().getFirst();
    RushReferralService.getInstance().updateSourceInfoOfChild(child.getUserId(), parent.getUserId());
    asserts.assertRafFunctioningForFreeUser(parent, child, AutomationConstants.RafTypes.SIGNUP);
    RushUserService.getInstance().logoutUser(child);
    RushUserService.getInstance().logoutUser(parent);
}
```

```
@Test(groups = {"infra", "raf"})
public void verifyChildTurningVipForVipParent() throws Exception {
    deleteUsers = false;
    UserInfo child = RushReferralService.getInstance().createEligibleUserForRaf().getFirst();
    UserInfo parent = RushReferralService.getInstance().createEligibleUserForRaf().getFirst();
    RushUserService.getInstance().makeVIPMember(parent.getUserId(), makeVIPMember: true);
    RushReferralService.getInstance().updateSourceInfoOfChild(child.getUserId(), parent.getUserId());
    RushPaymentService.getInstance().initiatePayForPlans(child, AutomationConstants.PaymentPlanIds.VIP);
    asserts.assertRafFunctioningForVipUser(parent, child, AutomationConstants.RafTypes.VIP);
    RushUserService.getInstance().logoutUser(child);
    RushUserService.getInstance().logoutUser(parent);
}
```

```
public void assertSpecialWheelRewards(List<Spokes> spokesFromApi, AutomationConstants.SpecialWheelRewardType specialWheelRewardType){
    SpecialWheelRewards specialWheelRewards = MongoUtil.getInstance().getSpecialWheelRewards(specialWheelRewardType);
    if(specialWheelRewards.getSpokeData().getCash().getFreeUser() == null){
        int j = 0;
        for(Spoke spoke : specialWheelRewards.getSpokeList()){
            Assert.assertEquals(spoke.getRewardType(), spokesFromApi.get(j).getRewardType());
            Assert.assertEquals(spoke.getSpokeId(), spokesFromApi.get(j).getId());
            Assert.assertEquals(spoke.getType(), spokesFromApi.get(j).getType());
            Assert.assertEquals((double) spoke.getValue(), spokesFromApi.get(j).getValue());
            j++;
        }
        Assert.assertEquals(specialWheelRewards.getSpokeList().size(), j);
    }
    else{
        int j = 0;
        for(FreeUser freeUser : specialWheelRewards.getSpokeData().getCash().getFreeUser()){
            Assert.assertEquals(freeUser.getRewardType(), spokesFromApi.get(j).getRewardType());
            Assert.assertEquals(freeUser.getSpokeId(), spokesFromApi.get(j).getId());
            Assert.assertEquals(freeUser.getType(), spokesFromApi.get(j).getType());
            Assert.assertEquals((double) freeUser.getValue(), spokesFromApi.get(j).getValue());
            j++;
        }
        Assert.assertEquals(specialWheelRewards.getSpokeData().getCash().getFreeUser().size(), j);
    }
}
```

## Utility functions to reports logs while running test cases

```
public void testLog(String message) { reportLog(message, classLogger: null, logStatus: null, isTestLog: true, isReportLog: false, status: nu

public void infoLog(String message, Logger classLogger) {
    reportLog(message, classLogger, LogStatus.INFO, isTestLog: false, isReportLog: false, status: null);
}

public void errorLog(String message, Logger classLogger) {
    reportLog(message, classLogger, LogStatus.ERROR, isTestLog: false, isReportLog: false, status: null);
}

public void reportTestLog(String message) { reportLog(message, classLogger: null, logStatus: null, isTestLog: true, isReportLog: true, Stat

public void reportTestLog(String message, Status status) { reportLog(message, classLogger: null, logStatus: null, isTestLog: true, isRepo

public void reportInfoLog(String message, Logger classLogger) {
    reportLog(message, classLogger, LogStatus.INFO, isTestLog: false, isReportLog: true, Status.INFO);
}

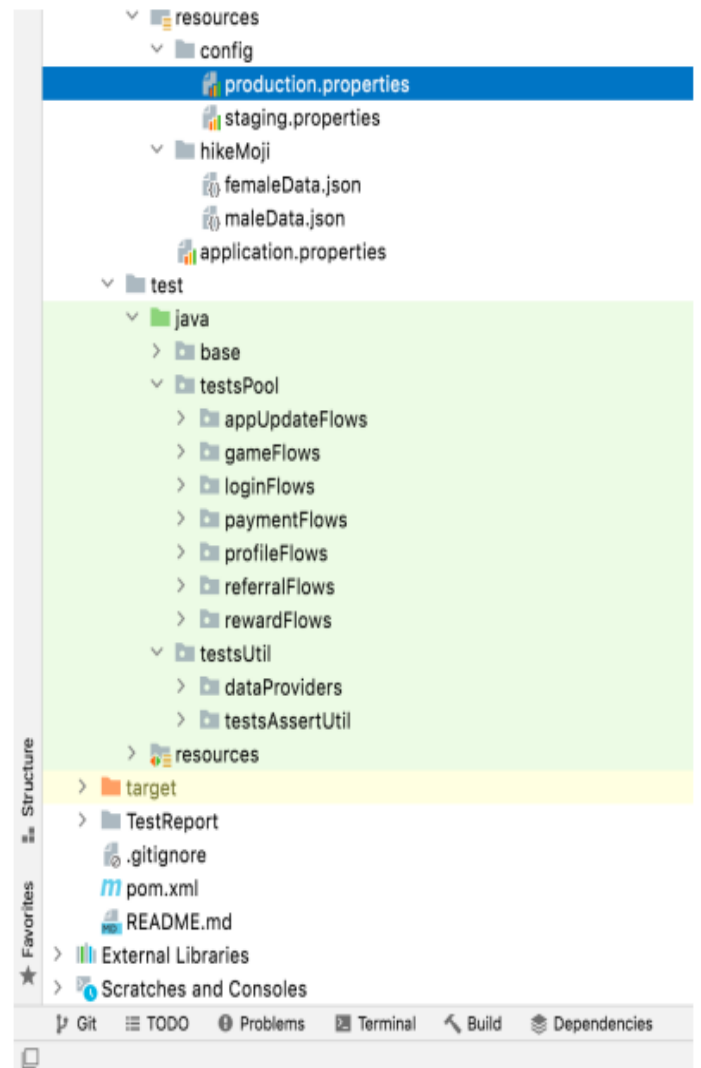
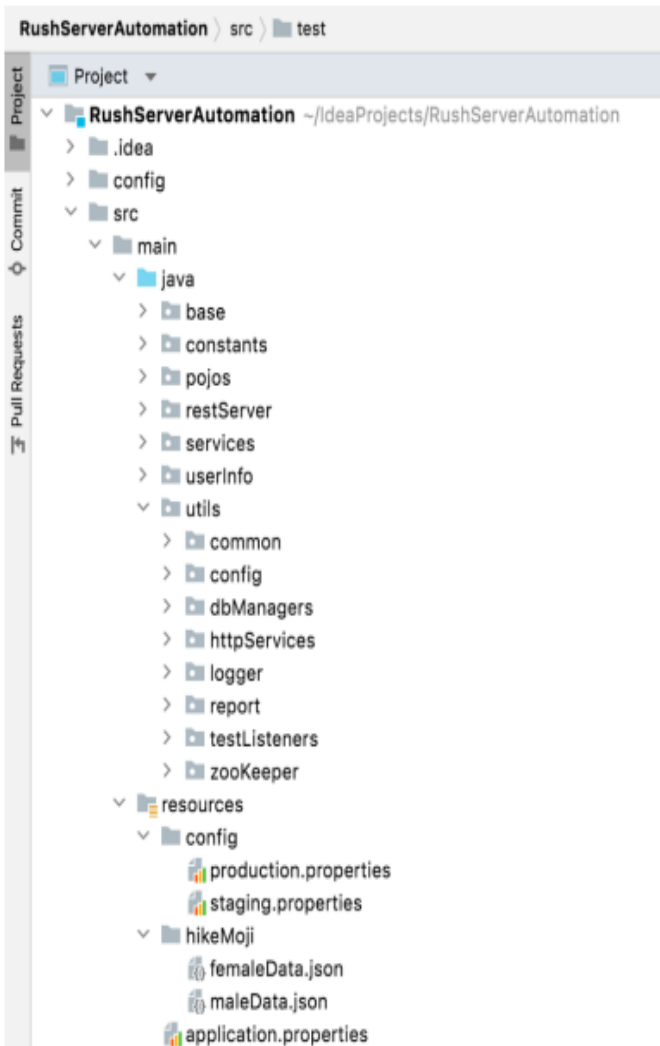
public void reportErrorLog(String message, Logger classLogger) {
    reportLog(message, classLogger, LogStatus.ERROR, isTestLog: false, isReportLog: true, Status.ERROR);
}

public void reportInfoLog(String message, Logger classLogger, boolean isTestLog, boolean isReportLog) {
    reportLog(message, classLogger, LogStatus.INFO, isTestLog, isReportLog, Status.INFO);
}
```

## Creating Report File of the Executed Cases

```
private static String getReportPath() {
    File testDirectory = new File(reportFilepath);
    reportFileName = reportFileName + "_" + TestListenerUtil.getTestSuiteName();
    DateTimeFormatter dtf = DateTimeFormatter.ofPattern("dd_MM_yy_HHmm");
    LocalDateTime now = LocalDateTime.now();
    reportFileLocation = reportFilepath + fileSeperator + reportFileName + "_" + dtf.format(now)
        + reportFileNameExt;
    if (!testDirectory.exists()) {
        if (testDirectory.mkdir()) {
            Reporter.log(s: "Directory: " + reportFilepath + " is created!", b: true);
            return reportFileLocation;
        } else {
            Reporter.log(s: "Failed to create directory: " + reportFilepath, b: true);
            return System.getProperty("user.dir");
        }
    } else {
        Reporter.log(s: "Directory already exists: " + reportFilepath, b: true);
    }
    return reportFileLocation;
}
```

# Project Structure





## **Conclusion**

According to an EY-All India Gaming Federation report, the online gaming sector in India is seen as one of the fastest-growing sectors, which is estimated to cross \$2 billion by 2023. Given the humongous potential that skill-based games hold for the future, especially towards enabling huge economic growth and in furthering 'Aatmanirbhar Bharat' over the long term, it is time India develops a clear, specific and effective policy making on this front to keep up the growth momentum of the skill gaming industry. At the same time, it is also upon all the stakeholders viz media, gamers and gaming enthusiasts and the gaming companies in our country to fight the misperception of skill based gaming with any type of gambling. Once there is a clear and government-authorised demarcation between skill-based games and chance-based games, it's quite likely that gaming would not have had to feel as much heat or wrath as it is facing today due to the perils associated with gambling.

## **Project Outcome**

Test Automation is a software testing technique that performs using special automated testing software tools to execute a test case suite. Test Automation is the best way to increase the effectiveness, test coverage, and execution speed in software testing. Test Tool selection largely depends on the technology the Application Under Test is built on. Test Automation Maintenance Approach is an automation testing phase carried out to test whether the new functionalities added to the software are working fine or not. The right selection of automation tool, testing process, and team, are important players for automation to be successful. Manual and automation methods go hand-in-hand for successful testing

## References

- i. [https://www.tutorialspoint.com/mongodb/mongodb\\_overview.htm](https://www.tutorialspoint.com/mongodb/mongodb_overview.htm)
- ii. <https://www.javatpoint.com/mysql-ql-tutorial>
- iii. [https://en.wikipedia.org/wiki/Charles\\_Proxy](https://en.wikipedia.org/wiki/Charles_Proxy)
- iv. <https://confluence.atlassian.com/bitbucketserver/basic-gitcommands-776639767.html>
- v. <https://www.ibm.com/analytics/hadoop/zookeeper>
- vi. <https://www.javatpoint.com/postman>
- vii. <https://www.indiatoday.in/impact-feature/story/how-indians-are-making-a-living-with-skill-based-games-1818095-2021-06-22>
- viii. <https://www.thehindubusinessline.com/opinion/skill-based-real-money-gaming-is-not-gambling/article36458914.ece>
- ix. <https://economictimes.indiatimes.com/industry/media/entertainment/skill-games-showed-hypergrowth-amid-lockdown-in-india/articleshow/83918887.cms?from=mdr>
- x. <https://qrius.com/the-changing-landscape-of-skill-gaming-in-india-an-in-depth-overview/>
- xi. <https://www.guru99.com/automation-testing.html>
- xii. <https://www.guru99.com/regression-testing.html>
- xiii. <https://www.guru99.com/integration-testing.html>
- xiv. <https://www.guru99.com/unit-testing-guide.html>

xv. [https://www.tutorialspoint.com/maven/maven\\_overview.html](https://www.tutorialspoint.com/maven/maven_overview.html)