CHAPTER 1
INTRODUCTION

1.1 Importance of software Testing

Software Testing is a process of identifying bugs or faults in your existing product before it is in the hand of our end users.

1). It's essential since it makes sure of the Customer's reliability and their satisfaction in the application.

2). It is very important to ensure the Quality of the product. Quality product delivered to the customers helps in gaining their confidence.

3). Testing is necessary in order to provide the facilities to the customers like the delivery of high quality product or software application which requires lower maintenance cost and hence results into more accurate, consistent and reliable results.

4). Testing is required for an effective performance of software application or product. 5). It's important to ensure that the application should not result into any failures because it can be very expensive in the future or in the later stages of the development.

6). It's required to stay in the business.

1.2 Testing Software

Software testing is an essential process conducted to identify defects, bugs, and errors in a software system. Its primary purpose is to ensure that the software functions as intended and meets the specified requirements. By testing the software, developers can find and fix issues before the software is released to end-users, thus preventing potential failures, malfunctions, or negative impacts on users. Testing also helps in enhancing the overall quality and reliability of the software by detecting and addressing defects. It plays a critical role in mitigating risks by identifying vulnerabilities and weaknesses that could lead to security breaches or performance issues. Moreover, testing validates changes and updates made to the software, ensuring that existing functionalities are not affected and that new features work as expected. Usability testing helps evaluate the software's user-friendliness and overall user experience, ensuring that it meets the needs and expectations of its intended users.

1.3 Advantages of software testing

1) Software testing helps in identifying and fixing bugs before the software becomes operational, the risk of failure can be reduced considerably.

2) Any software does not necessarily work alone. Sometimes it has to integrate and function with other existing legacy systems, as need be. In such cases software testing gives a much needed assurance that it will work suitably and its performance won't get affected due to the integration.

3) Software testing is a part of the software development process. It performs a root cause analysis for which helps in making it more efficient.

4) Software testing is carried out keeping the end user in mind. With a foresight for the scenarios that the end user will possibly face, tests are carried out and their accuracy and efficiency are duly noted for any discrepancies.

5) Reliability of software can be measured by certification or its conformation with any set technical standard. This helps in creating confidence and a healthy business relationship.

Software Testing can be of great use when done at right time as it enhance the efficiency of the products by keeping end user in mind.

1.4 List of software testing applications

The tools are divided into different categories as follows:

- Test Management tools
- Functional Testing Tools
- Load Testing Tools

Test Management tools
- TET (Test Environment Toolkit)
- TETware
- Test Manager
- RTH

Functional Testing Tools
- Selenium
- Soapui
- Watir
- HTTP::Recorder
- WatiN

- Canoo WebTest
- Webcorder
- Solex
- Imprimatur
- SAMIE
- Swete
- ITP
- WET
- WebInject
- Katalon Studio

Load Testing Tools

- Jmeter
- FunkLoad

1.4 List of various testing techniques used in different applications.

1.4.1 Functional testing types include:

- Unit testing
- Integration testing
- System testing
- Sanity testing
- Smoke testing
- Interface testing
- Regression testing
- Beta/Acceptance testing

1.4.2 Non-functional testing types include:

- Performance Testing
- Load testing
- Stress testing
- Volume testing
- Security testing
- Compatibility testing
- Install testing

CHAPTER 2

ATTENDANCE MANAGEMENT SYSTEM

2.1 Introduction

The project "**Attendance Management System**" is implemented to reduce the manual work and enhances the accuracy of work in a restaurant. This project is also designed with full consideration tohelp the Faculty in an easy manner without any unnecessary wastage of time. This system can be implemented in Institutions. Online Attendance management system deals with the maintenance of the student's attendancedetails. It generates the attendance of the student on basis of presence in class. It is maintained on the daily basis of all particular class of their attendance. Every single staffs will be provided with the separate **Email &password** to make the respective class's student's status also more than One Classis Provided to take Attendance. The staffs handling the particular subjects responsible to make the attendance for all students. Only if the student present on that particular period, the attendance will be calculated. The students attendance reports based on every particular Subject will be consolidate and Bar Graphs will be generated.

2.2 Advantages of selected application in testing environment.

Testing a student management system application offers several advantages, including:

Bug Detection: Testing helps identify and uncover any defects or bugs in the application. By testing different functionalities and scenarios, one can ensure that the system works as expected and that any issues are discovered and addressed before the application is deployed for use.

Accurate Testing Scenarios: In a testing environment, it is crucial to replicate real-life scenarios accurately. A payroll management system allows testers to generate and manipulate realistic test data related to employee salaries, taxes, deductions, and other payroll components. This ensures that the testing scenarios closely resemble actual payroll processes.

Data Integrity and Consistency: Testing a payroll system often involves working with a large volume of data, including employee records, salary structures, and tax regulations. A payroll management system ensures data integrity and consistency by storing all relevant information in a centralized database. Testers can rely on this consistent data source for conducting accurate and reliable tests.

Test Case Reusability: Once test cases related to payroll management are designed and validated, a payroll management system allows for their reusability. Testers can save and reuse test cases for future testing cycles or when implementing updates or enhancements to the system. This saves time and effort, ensuring a consistent and efficient testing process.

Usability and User Experience: Testing allows you to evaluate the usability and user experience of the application. By conducting usability tests and gathering feedback from users, you can identify areas where the application can be improved to provide a better user experience and meet the needs of administrators, teachers, students, and parents

2.3 Testing techniques used in Attendance Management System

1. Unit testing

Unit testing is a level of software testing where individual units/ components of a software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of any software. It usually has one or a few inputs and usually a single output. In procedural programming, a unit may be an individual program, function, procedure, etc. In object-oriented programming, the smallest unit is a method, which may belong to a base/ super class, abstract class or derived/ child class. (Some treat a module of an application as a unit. This is to be discouraged as there will probably be many individual units within that module.) Unit testing frameworks, drivers, stubs, and mock/ fake objects are used to assist in unit testing.

2. Integration testing

Integration testing is a level of software testing where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units. Test drivers and test stubs are used to assist in Integration Testing. Approaches Big Bang is an approach to Integration Testing where all or most of the units are combined together and tested at one go. This approach is taken when the testing team receives the entire software in a bundle. So what is the difference between Big Bang Integration Testing and System Testing? Well, the former tests only the interactions between the units while the latter tests the entire system.

Top Down is an approach to Integration Testing where top-level units are tested first and lower level units are tested step by step after that. This approach is taken when top-down development

approach is followed. Test Stubs are needed to simulate lower level units which may not be available during the initial phases.

Bottom Up is an approach to Integration Testing where bottom level units are tested first and upper-level units step by step after that. This approach is taken when bottom-up development approach is followed. Test Drivers are needed to simulate higher level units which may not be available during the initial phases.

Sandwich/Hybrid is an approach to Integration Testing which is a combination of Top Down and Bottom Up approaches.

3. System testing

System testing is testing conducted on a complete integrated system to evaluate the system's compliance with its specified requirements.

System testing takes, as its input, all of the integrated components that have passed integration testing. The purpose of integration testing is to detect any inconsistencies between the units that are integrated together (called assemblages). System testing seeks to detect defects both within the "inter-assemblages" and also within the system as a whole. The actual result is the behaviour produced or observed when a component or system is tested.

System testing is performed on the entire system in the context of either functional requirement specifications (FRS) or system requirement specification (SRS), or both. System testing tests not only the design, but also the behaviour and even the believed expectations of the customer. It is also intended to test up to and beyond the bounds defined in the software or hardware requirements specification(s).

4. Boundary value testing

Boundary testing is the process of testing between extreme ends or boundaries between partitions of the input values. So these extreme ends like Start- End, Lower- Upper, Maximum-Minimum, Just Inside-Just Outside values are called boundary values and the testing is called "boundary testing".

The basic idea in boundary value testing is to select input variable values at their:

1. Minimum
2. Just above the minimum
3. A nominal value
4. Just below the maximum +-
5. Maximum

5. Equivalence class testing

Equivalent Class Partitioning is a black box technique (code is not visible to tester) which can be applied to all levels of testing like unit, integration, system, etc. Equivalence partitioning or equivalence class partitioning (ECP) is a software testing technique that divides the input data of a software unit into partitions of equivalent data from which test cases can be derived. In principle, test cases are designed to cover each partition at least once. This technique tries to define test cases that uncover classes of errors, thereby reducing the total number of test cases that must be developed. An advantage of this approach is reduction in the time required for testing software due to lesser number of test cases.

Equivalence partitioning is typically applied to the inputs of a tested component, but may be applied to the outputs in rare cases. The equivalence partitions are usually derived from the requirements specification for input attributes that influence the processing of the test object.

2.4   Pseudo code

```php
<?php session_start(); ?>
<?php include 'node_class.php'; ?>
<?php
  $teacher_id = $_SESSION['teacher_id'];
  $code = strtoupper($_POST['code']);
  $year = $_POST['year'];
  $semester = $_POST['semester'];
  $section = $_POST['section'];
  $startRoll = strtoupper($_POST['start']);
  $endRoll = strtoupper($_POST['end']);
  $classes = $_SESSION['classes'] == 0?array():$_SESSION['classes'];

  if(verify(CODE,$code) === false) respond("error","code");
  if(verify(ROLL,$startRoll) === false) respond("error","roll");
  if(verify(ROLL,$endRoll) === false) respond("error","roll");
  if(verify(NUMBER,$semester) === false) respond("error","semester");
  if(verify(NUMBER,$section) === false) respond("error","section");
  if(verify(NUMBER,$year) === false) respond("error","year");

  $n = new Node($code,$teacher_id,$year,$semester,$section,$startRoll,$endRoll) or
respond("error","exists");
  updateSession($_SESSION['email']);
  $classes2 = $_SESSION['classes'];
  $class_id;
  foreach($classes2 as $c) {
    if(!in_array($c,$classes)) $class_id = $c;
```

```php
 }
 if(!isset($class_id)) respond("error","exists");
 echo
json_encode(array("code"=>$code,"section"=>$section,"year"=>$year,"class_id"=>$class_id));
?>
<?php
DEFINE('DB_USER','root');
DEFINE('DB_PASS','');
DEFINE('DB_HOST','localhost');
DEFINE('DB_DB','attendance');
DEFINE('EMAIL',1);
DEFINE('PHONE',2);
DEFINE('ROLL',3);
DEFINE('CODE',4);
DEFINE('NAME',5);
DEFINE('NUMBER',6);


function connectTo() {
/*
 Does -> Connects to data base
 Returns -> Connection object
*/
 $con = new mysqli(DB_HOST,DB_USER,DB_PASS,DB_DB);
 return $con;
}
function sqlReady($input) {
/*
 Takes -> Any string
 Returns -> Escapes the string
*/
 $con = connectTo();
 $string = mysqli_real_escape_string($con,$input);
 $con->close();
 return $string;
}

function hashPass($pass,$rounds = 9) {
/*
 Takes -> Password
 Returns -> Hashes the password using blow-fish algorithm
*/
 $salt = "";
 $i = -1;
```

```php
  $saltChars = array_merge(range(0,9),range('a','z'),range('A','Z'));
  while(++$i < 22)
    $salt .= $saltChars[array_rand($saltChars)];
  return crypt($pass, sprintf('$2y$%02d$', $rounds) . $salt);
}
function verifyPass($input,$pass) {
/*
 Takes -> 2 Password strings
 Returns -> true if matches false if doesn't
*/
  return crypt($input,$pass) == $pass? true : false ;
}
function respond($as,$what) {
/*
 Takes -> key and value
 Does -> Dies by printing json_encoded array having the key and value
*/
  die(json_encode(array($as=>$what)));
}
function updateSession($email) {
/*
 Takes -> email
 Does -> Updates the SESSION variable as per the email
*/
  $con = connectTo();
  $exists = $con->query("select * from `attendance`.`teacher` where email = '$email'");
  $exists = $exists->fetch_assoc();
  $_SESSION['name'] = $exists['name'];
  $_SESSION['email'] = $exists['email'];
  $_SESSION['phone'] = $exists['phone'];
  $_SESSION['teacher_id'] = $exists['uid'];
  $_SESSION['classes'] = 0;
  $classes = $con->query('select uid from `objects` where teacher_uid =
'.$_SESSION['teacher_id']);
  if($classes && $con->affected_rows) {
    $cls = array();
    while($a = $classes->fetch_array()) {
      $cls[] = $a[0];
    }
    $_SESSION['classes'] = $cls;
  }
  $con->close();
  session_write_close();
}
```

```php
function verify($type,$input) {
/*
 Takes -> Type of regex checker and the input
 Does -> Computes the regex
 Returns -> Returns true and false
*/
 $reEmail = '/^([\S]+)@([\S]+)\.([\S]+)$/';
 $rePhone = '/^[0-9]{10}$/';
 $reCode  = '/^([a-zA-Z]{3})\-([0-9]{3})$/';
 $reRoll  = '/^([0-9]{3})\/([a-zA-z]{2})\/([0-9]{2})$/';
 $reName  = '/^[a-zA-Z \']+$/';
 $reNum   = '/^[0-9]+$/';
 $m;
 switch($type) {
   case EMAIL :
     preg_match($reEmail,$input,$m);
   break;
   case PHONE :
     preg_match($rePhone,$input,$m);
   break;
   case CODE :
     preg_match($reCode,$input,$m);
   break;
   case ROLL :
     preg_match($reRoll,$input,$m);
   break;
   case NAME :
     preg_match($reName,$input,$m);
   break;
   case NUMBER :
     preg_match($reNum,$input,$m);
   break;
 }
 return count($m) == 0? false : true;
}
?>
```

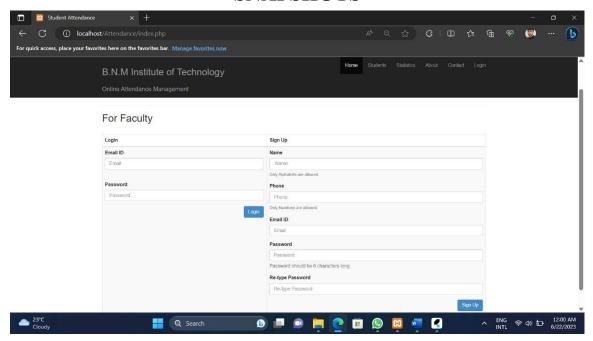# CHAPTER 3

SNAPSHOTS

Fig 3.1 Admin Login

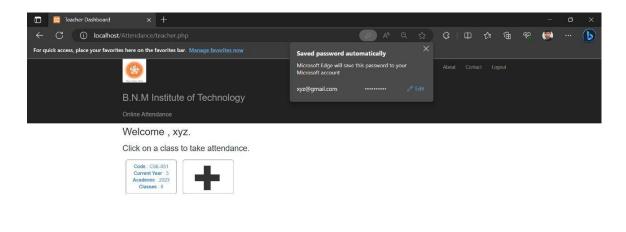This is the snapshot of Admin login page where admin can enter the username and password to login to the account.
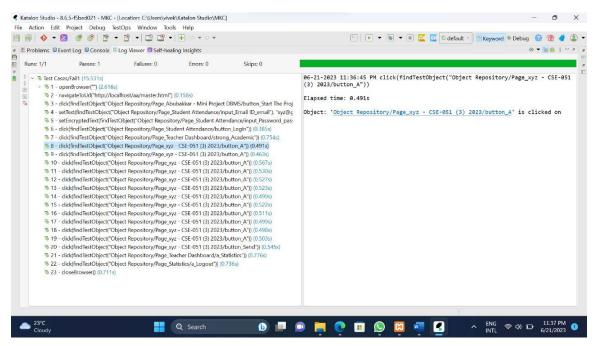

Fig 3.2 Landing Page

Fig 3.3 Log View

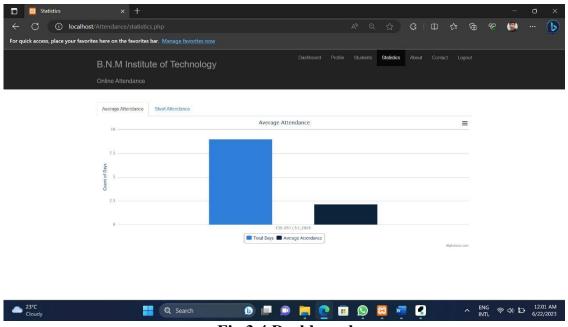This snapshot displayed elapsed time for each clip.

Fig 3.4 Dashboard

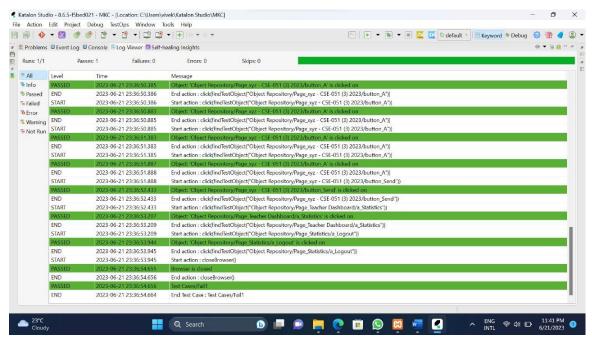This snapshot displays statistics of class attendance.

Fig 3.5 Log View

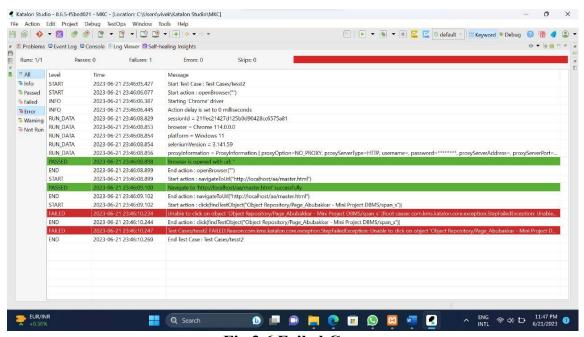This snapshot displays tabular form of all passed test cases.

Fig 3.6 Failed Case

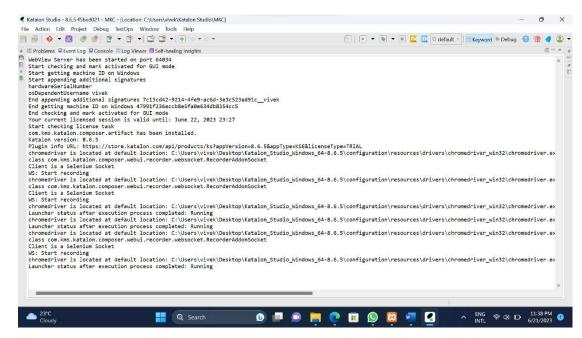This snapshot displays tabular form of failed test cases.

Fig 3.7 Event log

This snapshot displays event log of the test case

CHAPTER 4
TEST EXECUTION PLAN

4.1 Test Strategy

Revision History

Date	Version	Author	Description
21-06-2023	1.0	Sahil, Tushar, Vivek	Test case with satisfying conditions
21-06-2023	1.1	Sahil, Tushar, Vivek	Test case to check failure

4.2 Scope:

Katalon Studio follows the Page Object Model pattern. GUI elements and API methods can be captured using the recording utility and stored into the Object Repository which is accessible and reusable across different test cases.

4.3 Test Environment:

Apache HTTP Server:

The Apache HTTP Server, colloquially called Apache is free and open sources crossplatform web server software, released under the terms of Apache License 2.0. Apache is developed and maintained by an open community of developers under the auspices of the Apache Software Foundation. So basically, a web server is the software that receives your request to access a web page. It runs a few security checks on your HTTP request and takes you to the web page. Depending on the page you have requested, the page may ask the server to run a few extra modules while generating the document to serve you. It then serves you the document you requested. Apache is the most popular Web server software. It enables a computer to host one or more Websites that can be accessed over the Internet using a Web Browser. The first version of Apache was released in 1995 by the Apache Group. In 1999, the Apache Group became the Apache Software Foundation, a non-profit organization that currently maintains the development of the Apache Web server software. Apache's popularity in the Web hosting market is largely because it is open source and free to use. Therefore, Web hosting companies can offer Apache-based Web hosting solutions at minimal costs. Other server software, such as Windows Server, requires a commercial license. Apache also supports multiple platforms,

including Linux, Windows, and Macintosh operating systems. Since many Linux distributions are also open-source, the Linux/Apache combination has become the most popular Web hosting configuration. Apache can host static websites, as well as dynamic websites that use server-side scripting languages, such as PHP, Python, or Perl. Support for these and other languages is implemented through modules, or installation packages that are added to the standard Apache installation.

PHP:

PHP is an open-source, interpreted, and object-oriented scripting language that can be executed at the server-side. PHP is well suited for web development. Therefore, it is used to develop web applications (an application that executes on the server and generates the dynamic page.). PHP was created by Rasmus Lerdorf in 1994 but appeared in the market in 1995. PHP7.4.0isthe latest version of PHP, which was released on 28 November. Some important points need to be noticed about PHP are as followed: PHP stands for Hypertext Preprocessor. PHP is an interpreted language, i.e., there is no need for compilation. PHP is faster than other scripting languages, for example, ASP and JSP.PHP is a server-side scripting language, which is used to manage the dynamic content of the website.PHP can be embedded into HTML.PHP is an object-oriented language. PHP is an open-source scripting language. PHP is simple and easy.

Visual Studio Code:

Visual Studio Code is a lightweight but powerful source code editor which runs on your desktop and is available for Windows, macOS and Linux. It comes with built-in support for JavaScript, Type Script and Node.js and has a rich ecosystem of extensions for other languages and runtimes (such as C++, C#, Java, Python, PHP, Go, .NET).

4.4 Risk Analysis:

- Possibility of application crush, if huge number of requests are sent.
- Application fails without the apache server running continuously in backend.
- Application may fail if any malware, virus or bug is detected.