Created By: Abhinab Talukdar
Date: 24th March,2024

Test Plan (KodNest Student Learning Application)

Test Plan 1

Objective 2

Scope 2

Inclusions 3

Test Environment 4

Defect Reporting Procedure 5

Test Strategy 5

Test Schedule 6

Test Deliverables 6

Entry and Exit Criteria 7

Entry Criteria 7

Exit Criteria 7

Test Execution 7

Entry Criteria 7

Exit Criteria 7

Test Closure 7

Entry Criteria 7

Exit Criteria 7

Tools 8

Risks and Mitigations 8

Approvals 8

Created By: Abhinab Talukdar
Date: 24th March,2024

Objective

In this document of the Test Plan for the **KodNest Student Learning Application** & measure its impact everywhere.

Import and aggregate all metrics you care about and measure how they get impacted by your experiments.

Track both leading and lagging indicators for your experiment's impact

Keep an eye on guardrail metrics to stop bad experiments early on

- React 18.2.0
- jQuery 2.1.1
- JavaScript
- Database SQL
- Web Server (Apache suggested)
- Nginx

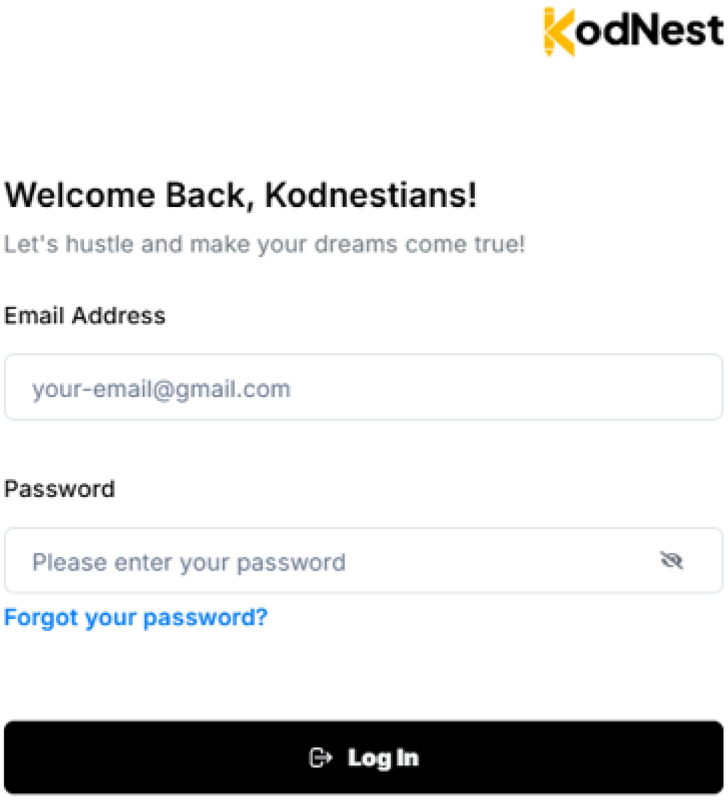
Scope

The end to end features and functionality of the *Student Learning Application* will be tested, such as the user **login, student dashboard, Course Management, Video Playback**, **assessment** and **evaluation, Mock Interview setup** and **Mark Attendance**.

The types of testing that will be performed, such as **Functional testing, Integration testing, performance testing,** and **UI testing**.

The environments in which testing will be conducted, such as different browsers, operating systems, and device types.

The criteria that will be used to evaluate the success of the testing, such as the number of defects found, the time taken to complete the testing, and user satisfaction ratings.



KodNest

Welcome Back, Kodnestians!

Let's hustle and make your dreams come true!

Email Address

your-email@gmail.com

Password

Please enter your password

The roles and responsibilities of the team members involved in the testing, such as the test lead, testers, and developers.

The schedule and milestones for the testing, including the start and end dates, and the planned testing activities.

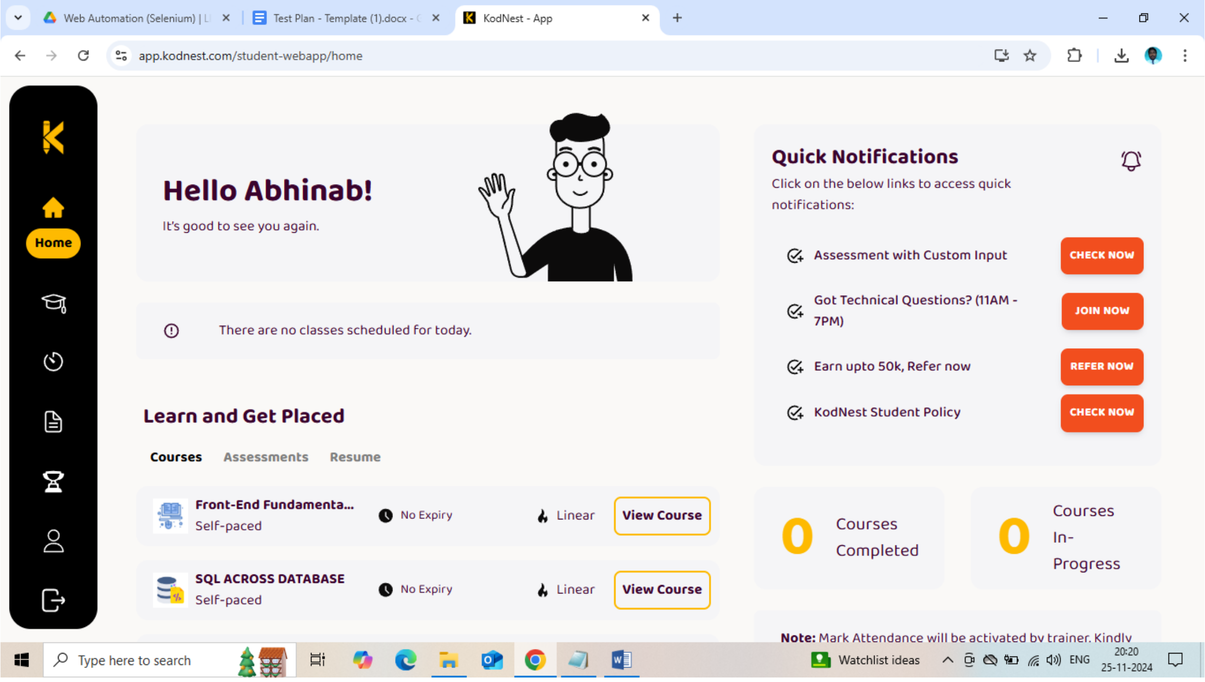
The tools and equipment that will be used for testing, such as testing software, hardware, and documentation templates.

Inclusions

Introduction: This section would provide an overview of the test plan, including its purpose, scope, and goals.

Test Objectives: This section would outline the specific objectives of the testing, such as identifying and fixing defects, improving the user experience, or achieving a certain level of performance.

- Login
- Dashboard Page
- Create Account



Hello Abhinab!

It's good to see you again.

There are no classes scheduled for today.

Learn and Get Placed

Front-End Fundamenta... Self-paced

SQL ACROSS DATABASE Self-paced

Quick Notifications

Click on the below links to access quick notifications:

Assessment with Custom Input

Got Technical Questions? (11AM - 7PM)

Earn upto 50k, Refer now

KodNest Student Policy

0 Courses Completed

0 Courses In-Progress

Note: Mark Attendance will be activated by trainer. Kindly

Exclusion

1. Support Page
2. Review page

Test Environments

The operating systems and version that will be used for testing, such as Windows 10.

The browsers and versions that will be tested, such as Google Chrome, Mozilla Firefox, or Microsoft Edge.

The device types and screen sizes that will be used for testing, such as desktop computers, laptops and smartphones.

The network connectivity and bandwidth that will be available for testing, such as Wi-Fi, cellular, or wired connections.

The hardware and software requirements for running the test cases, such as a specific processor, memory, or storage capacity.

The security protocols and authentication that will be used to access the test environment, such as passwords, tokens, or certificates.

Windows 10 – *Chrome, Firefox and Edge*

Android Mobile OS – *Chrome*

Defect Reporting Procedure

The criteria for identifying a defect, such as deviation from the requirements, user experience issues, or technical errors.

The steps for reporting a defect, such as using a designated template, providing detailed reproduction steps, and attaching screenshots or logs.

The process for triaging and prioritizing defects, such as assigning severity and priority levels, and assigning them to the appropriate team members for investigation and resolution.

The tools and systems that will be used for tracking and managing defects, such as a defect tracking software or a project management tool.

The roles and responsibilities of the team members involved in the defect reporting process, such as testers, developers, and the test lead.

The communication channels and frequencies for updating stakeholders on the progress and status of defects.

The metrics that will be used to measure the effectiveness of the defect reporting process, such as the number of defects found, the time taken to resolve them, and the percentage of defects that were successfully fixed.

Defect Process	**POC**
New FrontEnd	Abhinab
BackEnd	Akash
DevOps	Rajesh

Tools - JIRA

Test Strategy

Step 1: Create test scenarios and test cases for the various features in Scope.

While developing test cases, we'll use a number of test design techniques.

- Equivalence Class Partition
- Decision Table Testing
- State Transition Testing
- Use Case Testing

We also use our expertise in creating Test Cases by applying the below:

- Error Guessing
- Exploratory Testing
- We prioritize the Test Cases

Step 2: Our testing procedure when we receive a request for testing:

- First, we will conduct smoke tests to see if the various and important functionalities of the application are working.
- We reject the build if the Smoke Testing fails and will wait for the stable build before performing in depth testing of the application functionalities.
- Once we receive a stable build, which passes Smoke Testing, we perform in-depth testing of the test cases required.
- Multiple Test Resources will be testing the same Application on Multiple Supported Environments simultaneously.

We then report the bugs in bug tracking tool and send dev. Management the defect found on that day in a status end of the day email.

As part of the Testing, we will perform the below types of Testing:

- Smoke Testing and Sanity Testing
- Regression Testing and Retesting
- Usability Testing, Functionality & UI Testing
- We repeat Test Cycles until we get the quality product.

Step 3: We will follow the below best practices to make our Testing better:

- Context Driven Testing – We will be performing Testing as per the context of the given application.
- Shift Left Testing –We will start testing from the beginning stages of the Development itself, instead of waiting for the stable build.
- Exploratory Testing – Using our expertise we will perform Exploratory Testing, apart from the normal execution of the Test cases.
- End to End Flow Testing – We will test the end-to-end scenario which involve multiple functionalities to simulate the end user flows.

Test Schedule

Following is the test schedule planned for the project –

Task Time Duration

TASK	**DATES**
• Creating Test Plan	20th March,2024
• Test Case Creation	11th May,2024
• Test Case Execution	30th May,2024
• Summary Reports and Submission Dates	15th June,2024

Test Deliverables

Entry and Exit Criteria

The below are the entry and exit criteria for every phase of Software Testing Life Cycle:

Requirement Analysis

Entry Criteria:

- Once the testing team receives the Requirements Documents or details about the Project

Exit Criteria:

- List of Requirements are explored and understood by the Testing team
- Doubts are cleared

Test Execution

Entry Criteria:

- Test Scenarios and Test Cases Documents are signed-off by the Client
- Application is ready for Testing

Exit Criteria:

- Test Case Reports, Defect Reports are ready

Test Closure

Entry Criteria:

- Test Case Reports, Defect Reports are ready

Exit Criteria:

- Test Summary Reports

Tools

The following are the list of Tools we will be using in this Project:

- JIRA Bug Tracking Tool
- Mind map Tool
- Snipping Screenshot Tool
- Word and Excel documents

Risks and Mitigations

The following are the list of risks possible and the ways to mitigate them:

Risk: Non-Availability of a Resource

Mitigation: Backup Resource Planning

Risk: Build URL is not working

Mitigation: Resources will work on other tasks

Risk: Less time for Testing

Mitigation: Ramp up the resources based on the Client needs dynamically

Approvals

Team will send different types of documents for Client Approval like below:

- Test Plan
- Test Scenarios
- Test Cases
- Reports

Testing will only continue to the next steps once these approvals are done