

**MyLearnzia**

***Project Report***

******

# 

**Proposed By-**

**DIA  
 KUSH SAINI  
 RISHABH GUPTA  
 SANJUKTA GHOSH  
 SHIVANI SRIVASTAVA**

**Contents**

[**1. Introduction**](#_Toc43540721)

1.1 Overview……………………………………………………………………………………………………………………………………….....2

[1.2 Our Vision…………………………………………………………………………………………………………………………………………..2](#_Toc43540722)

[1.3 Purpose……………………………………………………………………………………………………………………………………………….2](#_Toc43540723)

[1.4 Abstract and use………………………………………………………………………………………………………………………………...2](#_Toc43540724)

1.5 Scope……………………………………………………………………………………………………………………………………………......2 1.6 Intended Audience…………………………………………………………………………………………………………………………......2

[2.Module Description](#_Toc43540725)

[2.1 Use case model 3](#_Toc43540726)

[2.1.1 Trainer Module use case ..3](#_Toc43540727)

[2.1.2Scholar Module use case ..4](#_Toc43540727)

[2.1.3Admin Module use case ..4](#_Toc43540727)

[3. Technical Specifications](#_Toc43540728)

[3.1 Technologies Used: .5](#_Toc43540729)

[3.2 Architecture Diagram .6](#_Toc43540730)

[3.2.1 Validation Architecture Diagram .6](#_Toc43540730)

[3.3 Database Design .7](#_Toc43540731)

3.4 [FlowCharts .7\](#_Toc43540732)

[4. Technical Detailed Description](#_Toc43540733)

[4.1 SDLC Layers……………………………………………………………………………………………………………………………………….14](#_Toc43540734)

[4.1.1 Application Implementation 15](#_Toc43540735)

[4.1.2 Application Architecture diagrams 16](#_Toc43540736)

[4 .2 Application/Business logic Layer……………………………………………………………………………………………………..17](#_Toc43540737)

[4.2.1 New Asset Request 18](#_Toc43540738)

[4.2.2 Asset Transfer Request 19](#_Toc43540739)

[4.3Data Layer………………………………………………………………………………………………………………………………………..20](#_Toc43540740)

**5. Risk Management**

6.1 Risk Identification…………………………………………………………………………………………………………………………21

6.2 Risk Mitigation………………………………………………………………………………………………………………………………21

6.3 Scheduling and estimates………………………………………………………………………………………………………………22

[6. Testing](#_Toc43540741)

[5.1 White-box testing 23](#_Toc43540742)

[5.2 Black-box testing 24](#_Toc43540743)

**7. Research and Team**

1. **INTRODUCTION**
   1. **OVERVIEW**

We are designing a web application named “***My Learnzia***”. It is a Learning Management System popularly known as LMS. We believe that the capacity to learn is a gift; the ability to learn is a skill; the willingness to learn is a choice.

Hence we bring in an application that will make learning easy and on your devices in our comfort zones. My Learnzia is a package as a whole that provides improved and better platform for the students to study.

* 1. **Our Vision**

We have chosen LMS as a project because we as scholars realize the importance of the learning management system and online learning. The whole idea behind the project is to fix the issues that that we have faced while using it. We are adding some   
important functionality as per our requirements. Moreover we are working on the user interface to make it a little attractive to study.

## Purpose

The purpose of this document is to define the requirements and design for a learning management system project  
 that can be used for the purpose of managing the learning or training in an organization. It contains the description   
 of all the modules, description and functionalities in detail.

## Abstract

The Learning Management System (LMS) is a web based application that is developed   
using the web technologies in JAVA, Spring Boot and Hibernate. The application is aimed to manage the learning and basic tasks related to learning of scholar in an organization. Based on the type of user it provides various required functionalities for Admin, scholar and Trainer. It facilitates administration, documentation, tracking, reporting, automation and delivery of training programs or learning and development programs.

## Scope

The LMS is a web application, with following functional requirements.

* 1. All the users will be able to login through same login page.
  2. User should be able to Change Password
  3. Admin can Create, Modify, View and Delete the users.
  4. The scholar will be able to access the content and phone directory, create and reply query, send emails and submit/view their assignments.
  5. Scholar should be able to view their query and its status.
  6. Scholar can view and edit their profile. They can also reply to queries created by other scholars and trainers.
  7. Trainer can View/Update/Reply the queries. Trainer can also mark attendance, send emails and access phone directory.
  8. The trainer can create/remove sessions, upload/edit/view assignment.
  9. Trainer can view their scholars and profile. They can also create/reply to queries.
  10. The Admin will insert/update/delete the calendar, study material, user profile, sessions, emails, queries.
  11. Admin should be able to approve the new request from scholar and should be able to monitor user activity and reports.
  12. **Intended Audience**

The main target o the application will be **Small to medium size training centres**, language schools, handcraft centres, ski academies, companies providing external training programs and organization that deliver effective and independent learning and development programs.

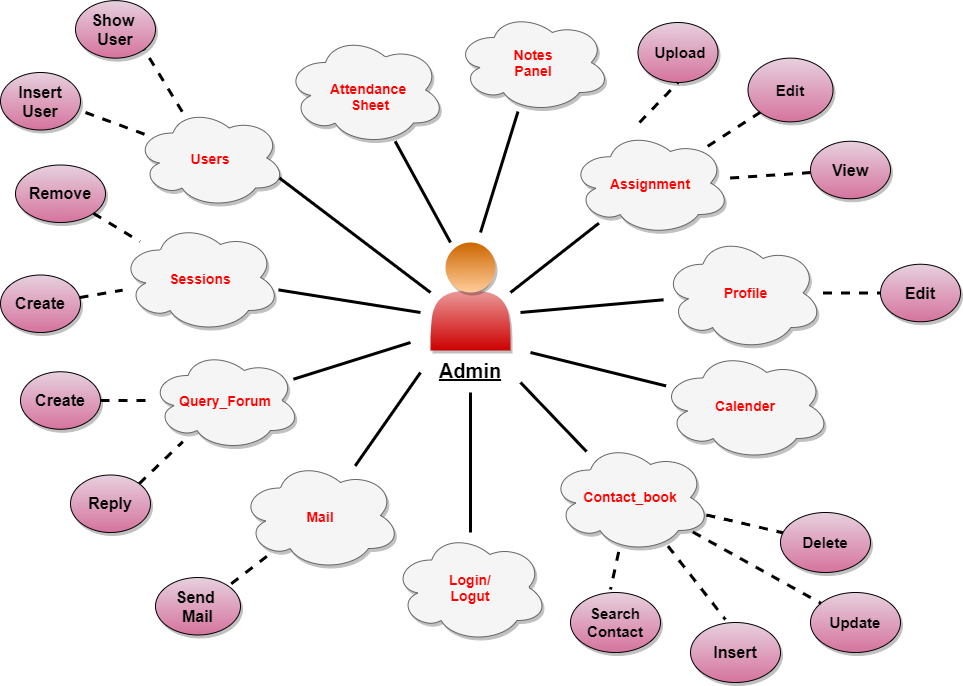
**2. Module Description**

2.1 Use case model

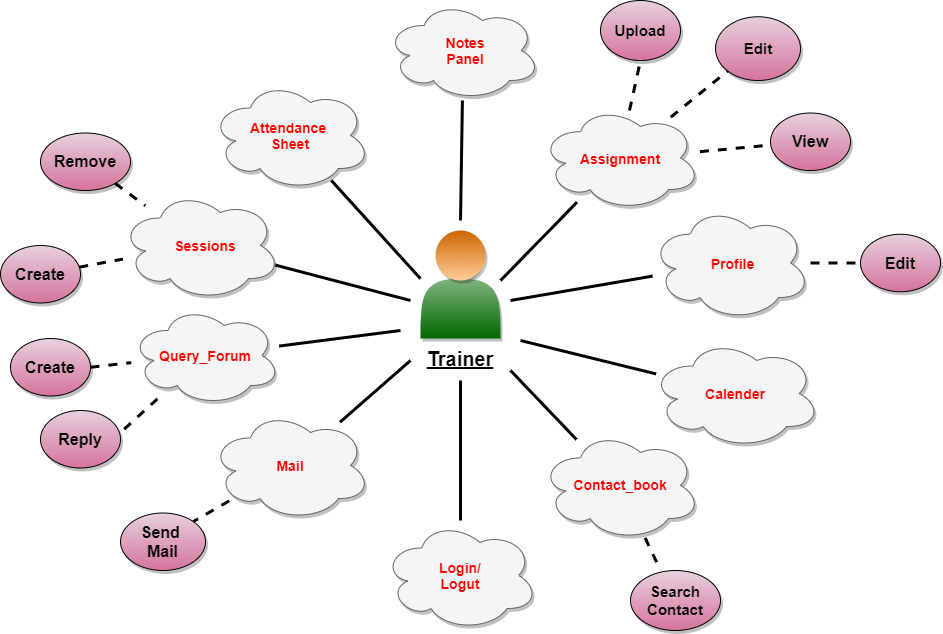
There are 3 roles in this Project. They are Admin, Trainer and Scholar.

* Admin Module
* Trainer Module
* Scholar Module

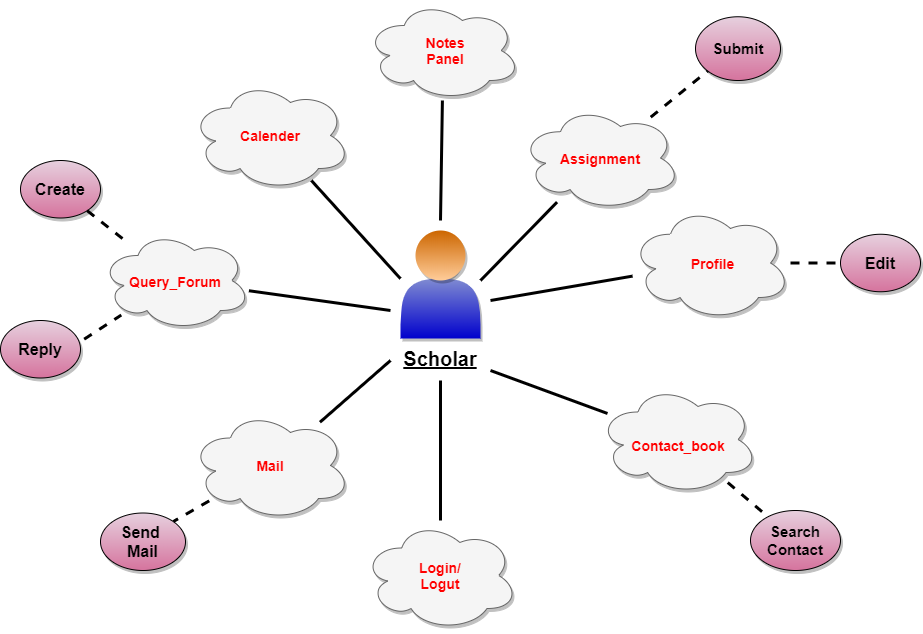
2.1.1 Admin Module Use Case



2.1.2 Trainer Module use case



2.1.3 Scholar Module use case



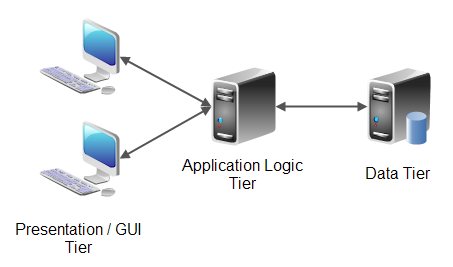
# 3. Technical Specifications

## 3.1 Technologies Used:

|  |  |
| --- | --- |
| Description | Software |
| Platform | Java |
| Database | Oracle Database |
| Server | Tomcat Server |

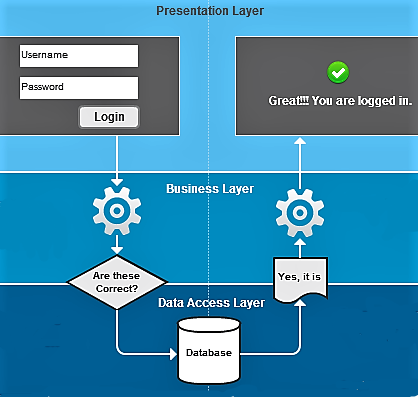
## 3.2 Architecture Diagram

The project is developed using 3-Tier Architecture. If we consider a 3-Tier Architectural WEB application, the browser becomes the client side application. The user communicates with the WEB/Application server via the browser. In the context of this project, the web pages represent the presentation layer. Here we can write code to design frontend and can perform validations. The data may be passed from this layer to business tier which has classes, methods and database connections. Now the queries are sent to Data Tier which gets data from Oracle Database and returns it to Business Tier.



## 3.2.1Validation Architecture Diagram

The project is following the same process for validation in the application by connecting through all the tier of architecture of the application.



## 3.3 Database Design

### Users\_Details:

|  |  |  |
| --- | --- | --- |
| Column name | Type | Size |
| Username | Varchar2 | 15 |
| Password | Varchar2 | 15 |
| Name | Varchar2 | 30 |
| DOB | Date |  |
| Gender | Varchar2 | 10 |
| E-Mail | Varchar2 | 20 |
| Mobile\_No | Varchar2 | 11 |
| Address | Varchar2 | 40 |
| Date-Of-Joining | Date |  |
| Passcode | Varchar2 | 20 |

### Query\_table:

|  |  |  |
| --- | --- | --- |
| Column name | Type | Size |
| Query\_id | int | 5 |
| Query | Varchar2 | 250 |
| Language\_query | Varchar2 | 20 |

### Feedback\_Table:

|  |  |  |
| --- | --- | --- |
| Column name | Type | Size |
| Trainer\_Name | Varchar2 | 30 |
| Feedback | Varchar2 | 255 |

### Quote\_Table:

|  |  |  |
| --- | --- | --- |
| Column name | Type | Size |
| Quote\_ID | Int | 5 |
| Quote\_data | Varchar2 | 256 |
|  |  |  |

### Session\_Table :

|  |  |  |
| --- | --- | --- |
| Column name | Type | Size |
| Session\_Id | Int | 5 |
| Session\_Title | TIMESTAMP | 50 |
| Start\_Time | TIMESTAMP |  |
| End\_Time | Varchar2 |  |
| Batch | Varchar2 | 25 |
| Venue | Varchar2 | 30 |
| Trainer | Trainer | 30 | 30 |

*Quote\_Table*

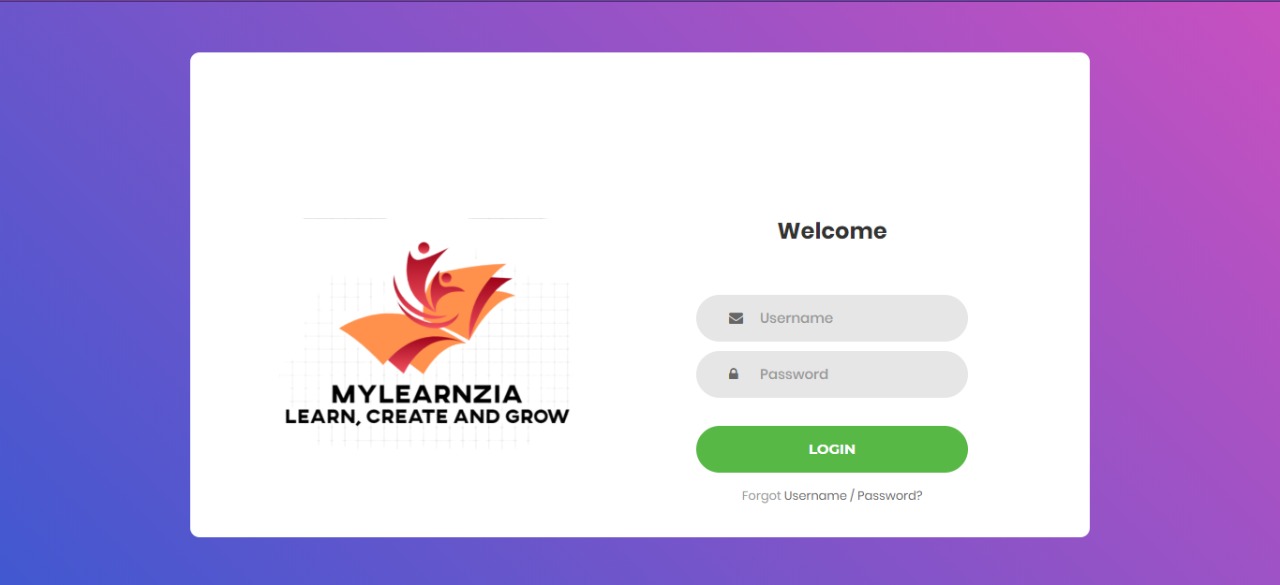
|  |  |  |
| --- | --- | --- |
| Column name | Type | Size |
| *Quote\_id* | *int* | *5* |
| *Quote\_data* | *Varchar2* | *256* |

4. Technical Specification

4.1. Presentation Layer

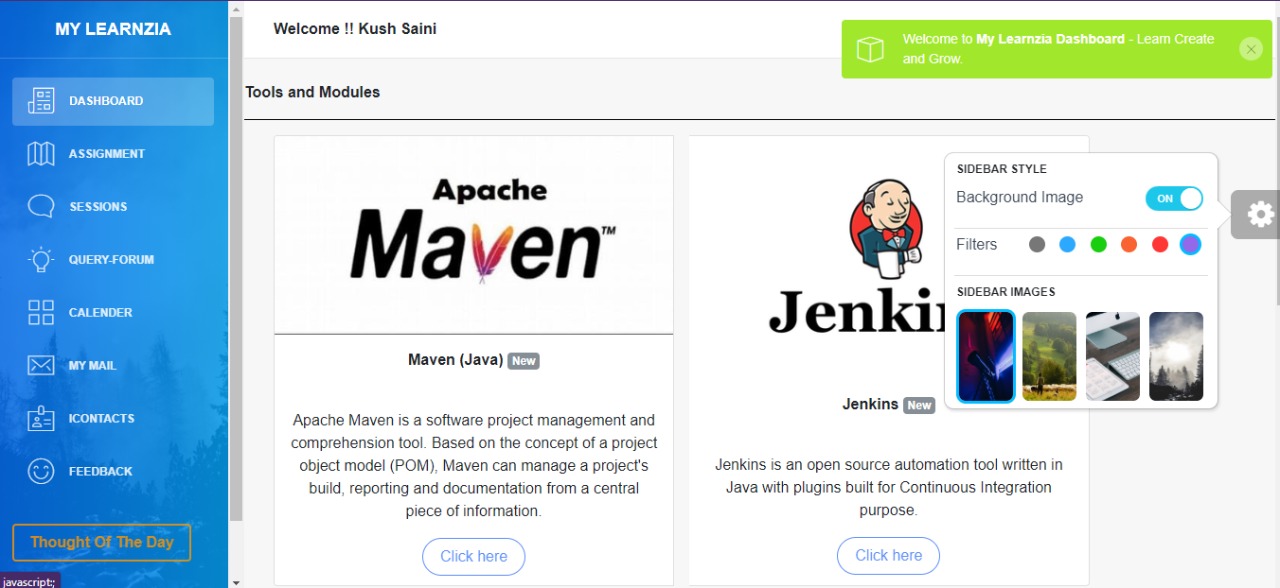
4.1.1. Login Page

Designed using Bootstrap and validation of details is done using database.



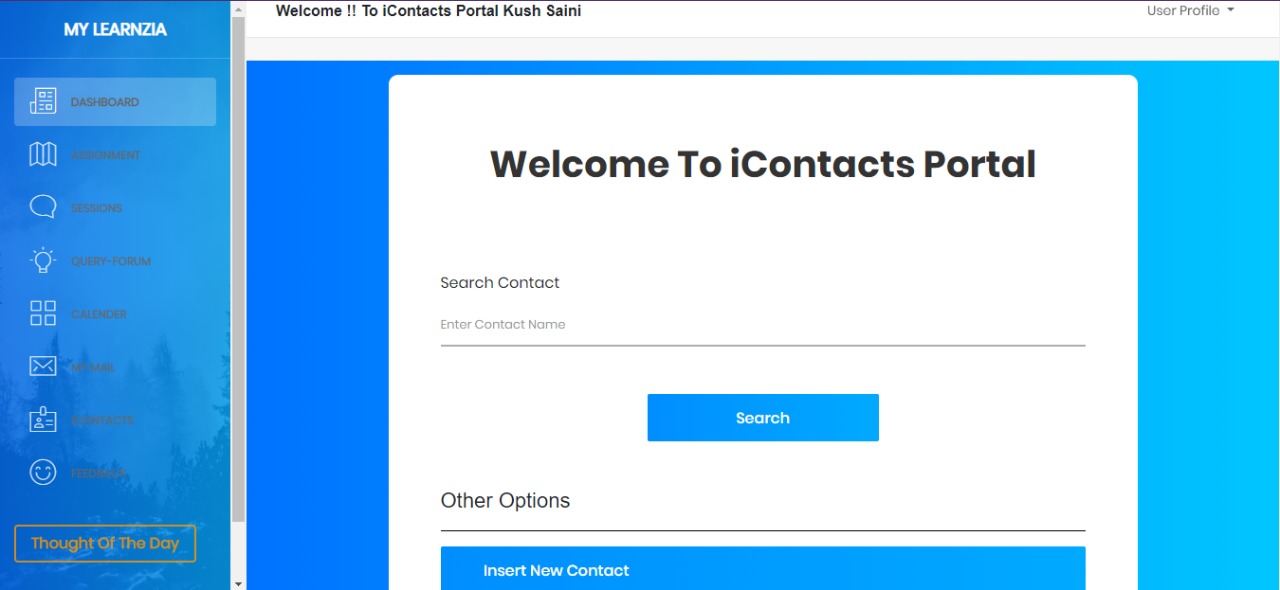
4.1.2. My Dashboard Page

It is designed using Bootstrap and details are added using database.



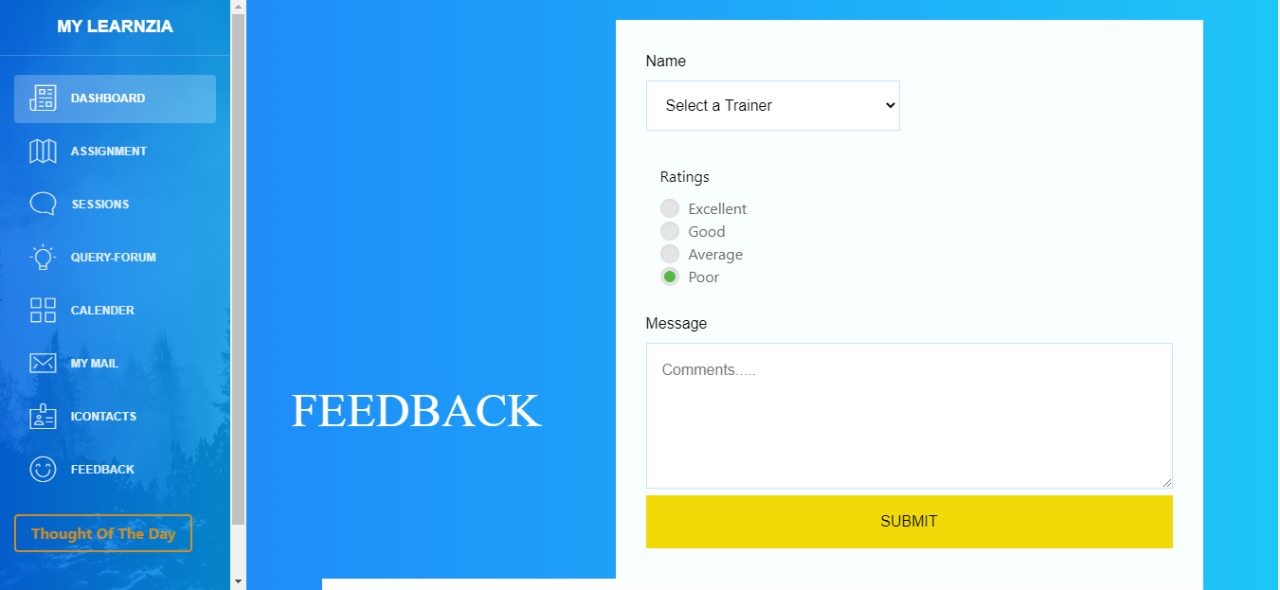
4.1.3. IContacts

It is designed using Bootstrap Ajax and Json.



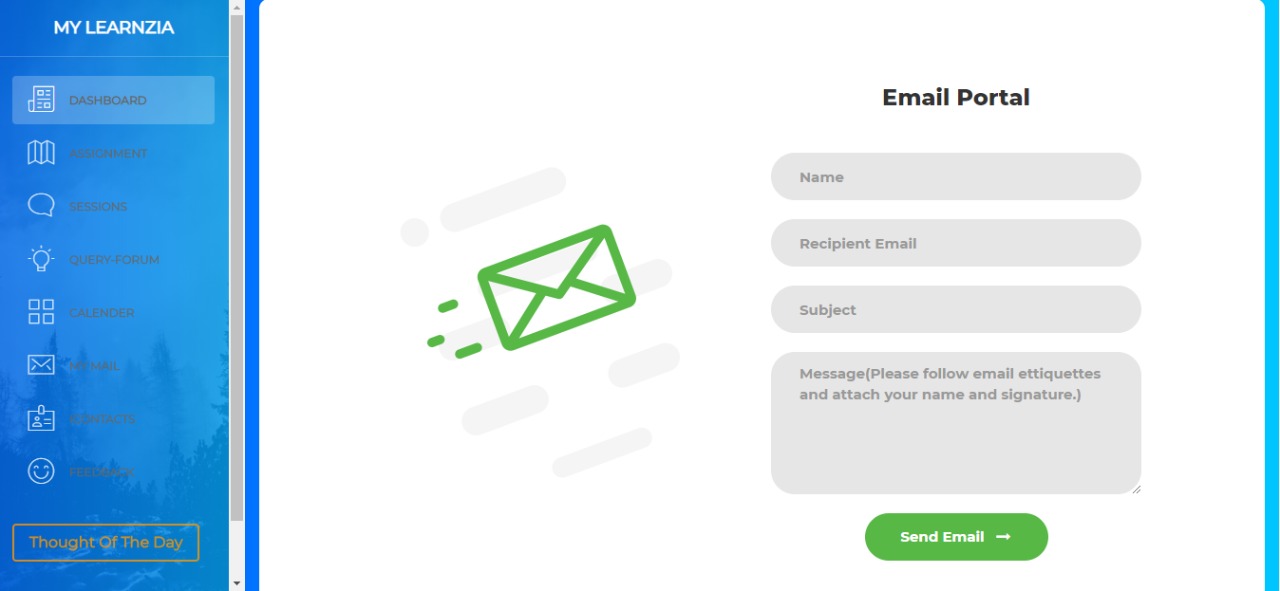
4.1.4. Feedback

It is designed using Bootstrap, tailwind,css and details are added using database.



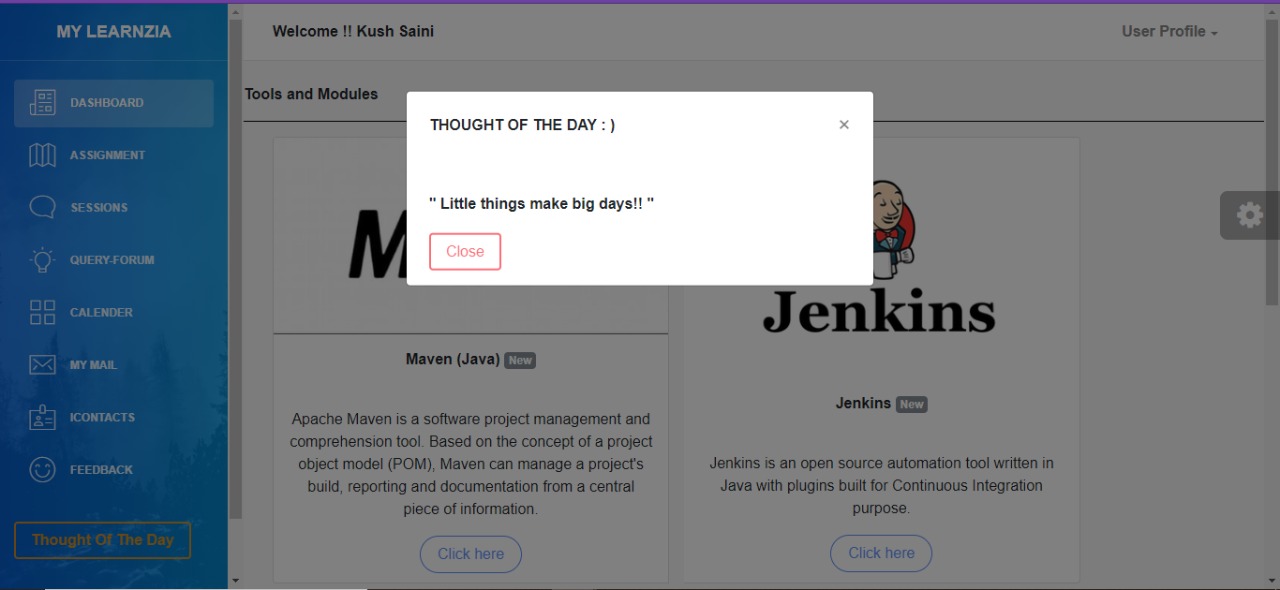
4.1.5. Email

It is designed using Bootstrap and SMTP protocol.



4.1.6. Thought of the day

It is designed using Bootstrap and details are added using database.



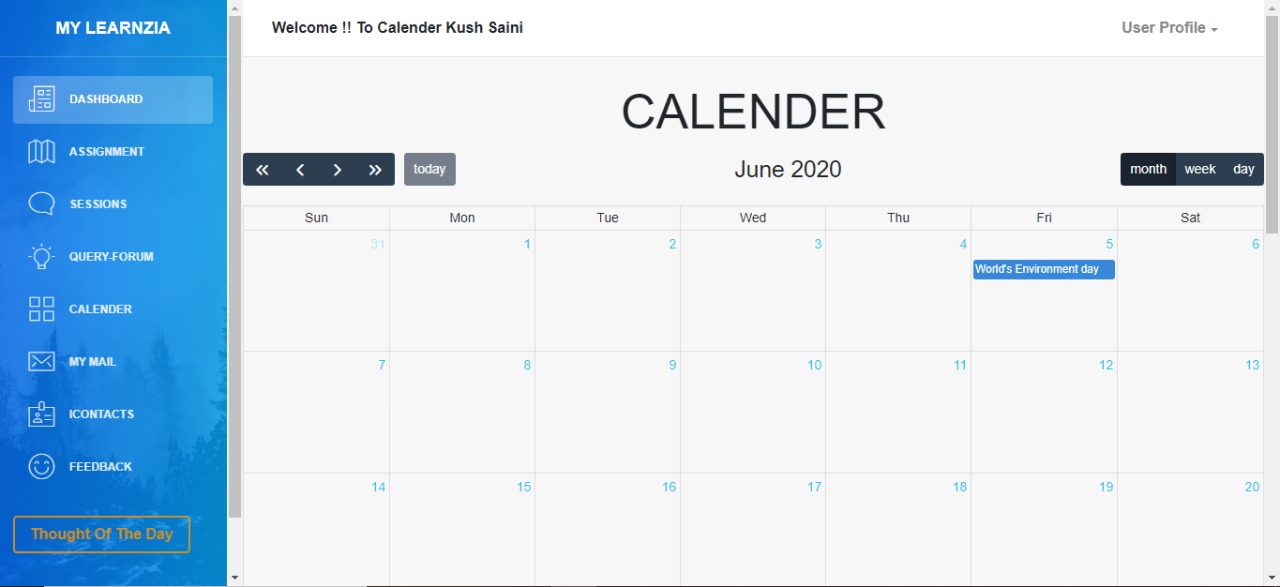
4.1.7. Sessions page

It is designed using Bootstrap and details are added using database.



4.1.8. Calendar

It is designed using Bootstrap and details are added using database.



***5. Risk Management***

***5.1. Risk Identification***

The major threats and as follows:

1. Confidentiality and integrity violation: unauthorized access and tempering the assets.
2. Denial of services: prevention of legitimate users.
3. Illegitimate use: Exploitation of privileges by legitimate users.
4. The risks can be unintended, deliberate or natural.

***5.2. Risk Mitigation***

The threats can be avoided by following some simple steps:

1. Allowing only authorized users as the username will be provided by the admin only.
2. The details will be feed in the database and the server will ensure that the access is given to legitimate users.
3. The admin will have access to all the activities and the privileges which will be monitored with he database.
4. The support team will always be available to take actions whenever needed.

# 6. TESTING

## 6.1. WHITE-BOX TESTING

White-box testing (also known as clear box testing, glass box testing and structural testing, by seeing the source code) tests internal structures or workings of a program, as opposed to the functionality exposed to the end-user.

This is done by the developer itself in parallel to developing of above project. This involves appropriate structuring of query and logic together.

## 6.2. BLACK-BOX-TESTING

Black-box testing treats the software as a "black box", examining functionality without any knowledge of internal implementation, without seeing the source code.

Some example is:

* A login screen.
* A scholar screen. (Read only)
* A trainer screen. (Read – Write)
* An administration screen. (All privileges)

Once the application is built the black box testing is done by creating some users and test with proper test case scenarios.The UI (User Interface) for these four screens are developed with HTML, JSP and BOOTSTRAP and the validations performed through JavaScript. Logic is in the Servlet, Hibernate and DAO (Data Access Object). DAO is a class for connecting to the database.

## 6.3. Manual Java Application Testing:

During manual JAVA testing, a tester prepares the test cases from the detailed design document and tries to cover every scenario and code snippet possible.

### #1) JAVA UNIT TESTING

[Unit testing is a type of testing](https://www.softwaretestinghelp.com/unit-testing/) wherein a user needs to test the smallest of the code snippets for accuracy, correctness and meeting the requirements.

The login screen has two text fields: username and password, one link to select your role (scholar or trainer), and has two buttons: login and cancel.

The test cases should cover all the loops and conditional statements. Test cases should display the expected results and the test data. Below are some of the general test cases that a user could execute manually in a login screen. The results are then noted down in the test case document.

**Below are the text cases:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.no | Test Case | Expected result | Actual Result | PASS/FAIL |
| 1. | User checks the appearance of labels Username, Password | The labels should be correctly spelled and displayed in Normal sized font | The labels are correctly spelled and displayed in Normal sized font. | PASS |
| 2. | User checks the appearance of the button  Submit and Cancel. | The buttons should be displayed with the correct name. | The buttons are displayed with the correct name. | PASS |
| 3. | If The user leaves any field Blank | Error message “It cannot be empty” should be displayed. | Error message in case field is left blank. | PASS |
| 4. | User enters username as “abcd” and password as “xxxx”. | Error message  “Invalid username password combination”  Should be displayed. | Error message in case field is left blank. | PASS |
| 5. | User enters a username of more than 10 characters. | Error message  “Username should not be more than 10 characters” should be displayed | Error message in case field is left blank. | PASS |
| 6. | On searching contacts we will get suggestions for the name. | We will enter the first name and get the list of all the names starting with that name. | We need to enter the full name to get the contact | FAIL |
| 7. | On clicking display all query we will see all the queries entered till date will be displayed | All entries are displayed. | All entries are displayed. | PASS |
| 8. | On entering the query in the query form and clicking insert the query will be inserted and added to display query. | Query will be inserted. | Query will be inserted. | PASS |
| 9. | You can send an email using the email id functionality. | Email sent successfully will be displayed on sending the mail. | Email sent successfully will be displayed on sending the mail. | PASS |
| 10. | On clicking thought of the day a pop-up with a quote will be displayed. | The pop up with a thought appears on clicking the button. | The pop up with a thought appears on clicking the button. | PASS |
| 11. | We can submit assignments in assignment portal. | Assignment submitted successfully will be displayed. | Assignment submitted successfully is displayed. | PASS |

**SOME MORE TEST CASES:**

* Check if NULLS are not allowed for username and password
* Check for any exception including NULL pointer exception
* Check if username/password is in the correct format
* Check if numbers are not allowed for username
* Check if special characters are not allowed in Username
* Check if the correct combination of Username and password are entered, then the application takes you to the next screen, i.e. employee information screen
* Check if the username entered is of correct length
* Check if the username text field allows only the maximum number of characters specified for that field
* Check if the password field if specified in the requirements is visible as \* while entering
* Check if passwords are case sensitive
* Check if username is not case sensitive
* Check if login page does not remember the username or password, even after exiting
* Check if the Submit and Cancel button work as per requirement
* If using the application first time, check if the username has permission to enter the application
* Delete a username/password combination from the database and check if the combination is not able to login again
* For all the above cases, check if the appropriate validation error messages are shown
* Check if the Labels and Buttons are in the right place on the screen and that they display the text correctly
* Check if the screen appearances are as per requirements
* Check if logging is performed for required actions

### #2) INTEGRATION TESTING

In Integration testing, individual modules are integrated and tested together for correctness.

**Here are some sample integration test cases for the Employee Application example:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.no | Test Case | Expected Result | Actual Result | PASS/ FAIL |
| 1. | Session in all the integrated screen if the user has logged in. | Sessions should not be same in all the other new integrated  Screens. | Sessions are not same in all the other new integrated  Screens. | PASS |
| 2. | Insert, update and delete functionality in all the screens. | Modules should be deleted, upgraded and inserted by only administration screen. | Modules are deleted, upgraded and inserted by only administration screen. | PASS |
| 3. | Header, footer, screen size and appearance is good in every device (bootstrap) during integration. | Header, footer, screen size and appearance is adjustable in any device. | Header, footer, screen size and appearance is adjustable in any device. | PASS |
| 4. | Check that when clicking on Submit buttons, the control is transferred to the next screen. | Next page should be opened in a new tab after clicking submit buttons. | Next page should be opened in a new tab after clicking submit buttons. | PASS |
| 5. | Check that when clicking on the Cancel button, the action performed is cancelled. | After clicking on the Cancel button, the action performed is cancelled. | After clicking on the Cancel button, the action performed is cancelled. | PASS |
| 6. | Check the flow of data, either Object, XML or Session from end to end. Check for correctness. | Data flow should be smooth without any error. | Data flow should be smooth without any error. | PASS |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Check the flow of data, either Object, XML or Session from end to end. Check for correctness. | Data flow should be smooth without any error. |  |  |
| 7. | If retrieve data back from the application. | Screens are able to download documents from the application (except scholar). | Screens are able to download documents from the application (except scholar). | PASS |

### #2) SYSTEM TESTING

In System testing, the entire application is tested for functionality and completeness with respect to the requirements.

**We have used selenium for automation testing.**

***OUR TEAM***

*** ***

**KUSH SAINI DIA**

**JAVA BATCH JAVA BATCH**

**NOIECP0719012NOIECP0719020**

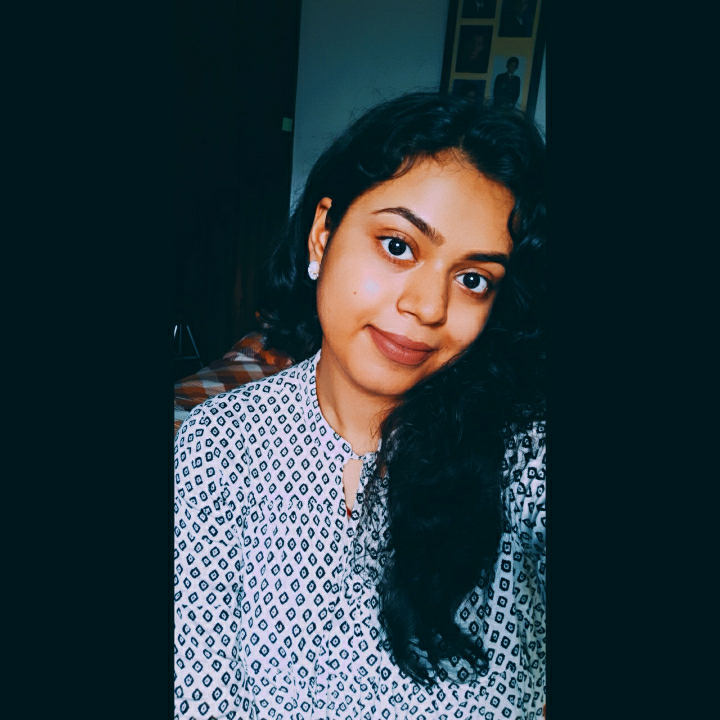
*** ***

Rishabh Gupta SANJUKTA GHOSH

JAVA BATCH JAVA BATCH

NOIECP0719031 NOIECP0719001

***OUR TEAM***

******

**SHIVANI SRIVASTAVA**

**JAVA BATCH**

**NOIECP0719015**

**(BHAGWAN KI MEHARBANI SE KRE HUE CODES)**

***BHAGWAN KHUSH HO GYE THE ISLIYE CHAL GYE***

***<3<3—from shivani***