

PROJECT DETAILS

PROJECT name - Phase 2 Full Stack Developer- Learners Academy
Developed By - Rohith V V
Trainer - Nisha Waiker

SPRINTS PLANNED AND TASKS ACHIEVED

Sprint Overview

Sprints

Sprint number	Code Name
1	Backend Development
2	WebView Creation
3	Testing and debugging
+	

✓ Tasks

▼ Tasks

Task	Sprint	Status	Assignee	Estimated Effort
create-a-basic-table-structure	Sprint 1	Done	rohith V V	6
create-the-hibernate-config	Sprint 1	Done	rohith V V	6
create-class-for-each-table	Sprint 1	Done	rohith V V	8
create-hbm.xml-files-for-the-classes	Sprint 1	Done	rohith V V	5
map-classes-to-config-file	Sprint 1	Done	rohith V V	6
create-Business-Object-interface-and-implement-it	Sprint 1	Done	rohith V V	10
create-DAO-interface-and-classes	Sprint 1	Done	rohith V V	13
create-a-login-page	Sprint 2	Done	rohith V V	9
create-the-dashboard	Sprint 2	Done	rohith V V	9
create-views-for-all-tableclasses	Sprint 2	Done	rohith V V	13
create-footer	Sprint 2	Done	rohith V V	5
create-necessary-custom-tags-and-implement-it	Sprint 2	Done	rohith V V	9
create-servlets-for-establishing-connection	Sprint 3	Done	rohith V V	13
manage-exceptions	Sprint 3	Done	rohith V V	13

ALGORITHM

1. ADMIN LOGIN

- Create a login page for entering credentials.
- Pass the credentials using session Tracking techniques.
- Check if the credentials are authentic.
- If it is authentic then go to next page where the menu is displayed.
- Else show that the credentials are not correct.

2. ADMIN MENU

- Show only if the credentials are true
- Display the options.
- Take user choice,
 - And as per the choice the corresponding operations are performed.

3. SETTING MASTER LIST OF STUDENTS

- Check if the database exists or not,
- If it exists,
 - Check for the student table.
 - if not present create one with required fields.
 - Take user inputs for the fields required .
 - Insert the user given values to the database..
- If it does not exists,
 - Display error message.

4. SETTING MASTER LIST OF TEACHERS

- Check if the database exists or not,
- If it exists,
 - Check for the teacher table.
 - if not present create one with required fields.

- Take user inputs for the fields required .
- Insert the user given values to the database..
- If it does not exists,
 - Display error message

5. SETTING MASTER LIST OF CLASSES

- Check if the database exists or not,
- If it exists,
 - Check for the classes table.
 - if not present create one with required fields.
 - Take user inputs for the fields required .
 - Insert the user given values to the database..
- If it does not exists,
 - Display error message

6. ASSIGNING CLASSES FOR SUBJECTS FROM MASTER LIST

- Create tables such that it follows many to one[may subject->one class]
- Show the master list of classes and subjects
- Take input for which subjects has to be assigned to which class.
- Store the resulting data in another table for convenience if required.

7. ASSIGNING TEACHERS TO CLASSES FROM MASTER LIST

- Create tables such that it follows one to many[one teacher->many class]
- Show the master list of classes and teachers.
- Take input for which teacher has to be assigned to which class.
- Store the resulting data in another table for convenience if required.

8. GETTING MASTER LIST OF STUDENTS

- Display the master list of students with fields such as class, subjects, roll number.

9. VIEWING CLASS REPORT

- Display all the data related to class such as
 - Teachers assigned,
 - Subjects,
 - Students name

CORE CONCEPTS USED

1. Hibernate.
2. Exception handling.
3. Servlets.
4. SQL.
5. JSP.
6. Regular Expressions.
7. Custom tags

LINK OF GITHUB REPOSITORY

https://github.com/Rohithvv98/RohithVV_Phase_2_Trainer_Nisha_Waikar.git

CONCLUSION

1. ENHANCING THE APPLICATION :add more options for the admin to interact with.
2. UNIQUE SELLING POINT(s) :errors are shown to admin in a clear manner, and redirect to login page.