simplilearn

JAVA-FSD (PHASE-2)

Developing a Backend Admin for Learner's Academy.

Developer details:-

Ritik

Technical Trainee

TEK System Global Services

Link to this project: https://github.com/Rkcr7/Admin-Portal_Phase2Project

PROJECT DESCRIPTION

Project objective:

As a Full Stack Developer, design and develop a backend administrative portal for the Learner's Academy. Use the GitHub repository to manage the project artifacts.

Background of the problem statement:

Learner's Academy is a school that has an online management system. The system keeps track of its classes, subjects, students, and teachers. It has a back-office application with a single administrator login.

The administrator can:

- Set up a master list of all the subjects for all the classes
- Set up a master list of all the teachers
- Set up a master list of all the classes
- Assign classes for subjects from the master list
- Assign teachers to a class for a subject (A teacher can be assigned to different classes for different subjects)
- Get a master list of students (Each student must be assigned to a single class)

There will be an option to view a Class Report which will show all the information about the class, such as the list of students, subjects, and teachers

The goal of the company is to deliver a high-end quality product as early as possible.

The flow and features of the application:

- Plan more than two sprints to complete the application
- Document the flow of the application and prepare a flow chart
- List the core concepts and algorithms being used to complete this application
- Implement the appropriate concepts, such as exceptions, collections, and sorting techniques for source code optimization and increased performance

You must use the following:

- Eclipse/IntelliJ: An IDE to code for the application
- Java: A programming language to develop the web pages, databases, and others
- SQL: To create tables for admin, classes, students, and other specifics
- Git: To connect and push files from the local system to GitHub
- GitHub: To store the application code and track its versions
- Scrum: An efficient agile framework to deliver the product incrementally
- Search and Sort techniques: Data structures used for the project
- Specification document: Any open-source document or Google Docs

The following requirements should be met:

- The source code should be pushed to your GitHub repository. You need to document the steps and write the algorithms in it.
- The submission of your GitHub repository link is mandatory. In order to track your task, you need to share the link of the repository. You can add a section in your document.
- Document the process step-by-step starting from sprint planning to the product release.
- The application should not close, exit, or throw an exception if the user specifies an invalid input.
- You need to submit the final specification document which will include:
- Project and developer details
- Sprints planned and the tasks achieved in them
- Algorithms and flowcharts of the application
- Core concepts used in the project
- Links to the GitHub repository to verify the project completion

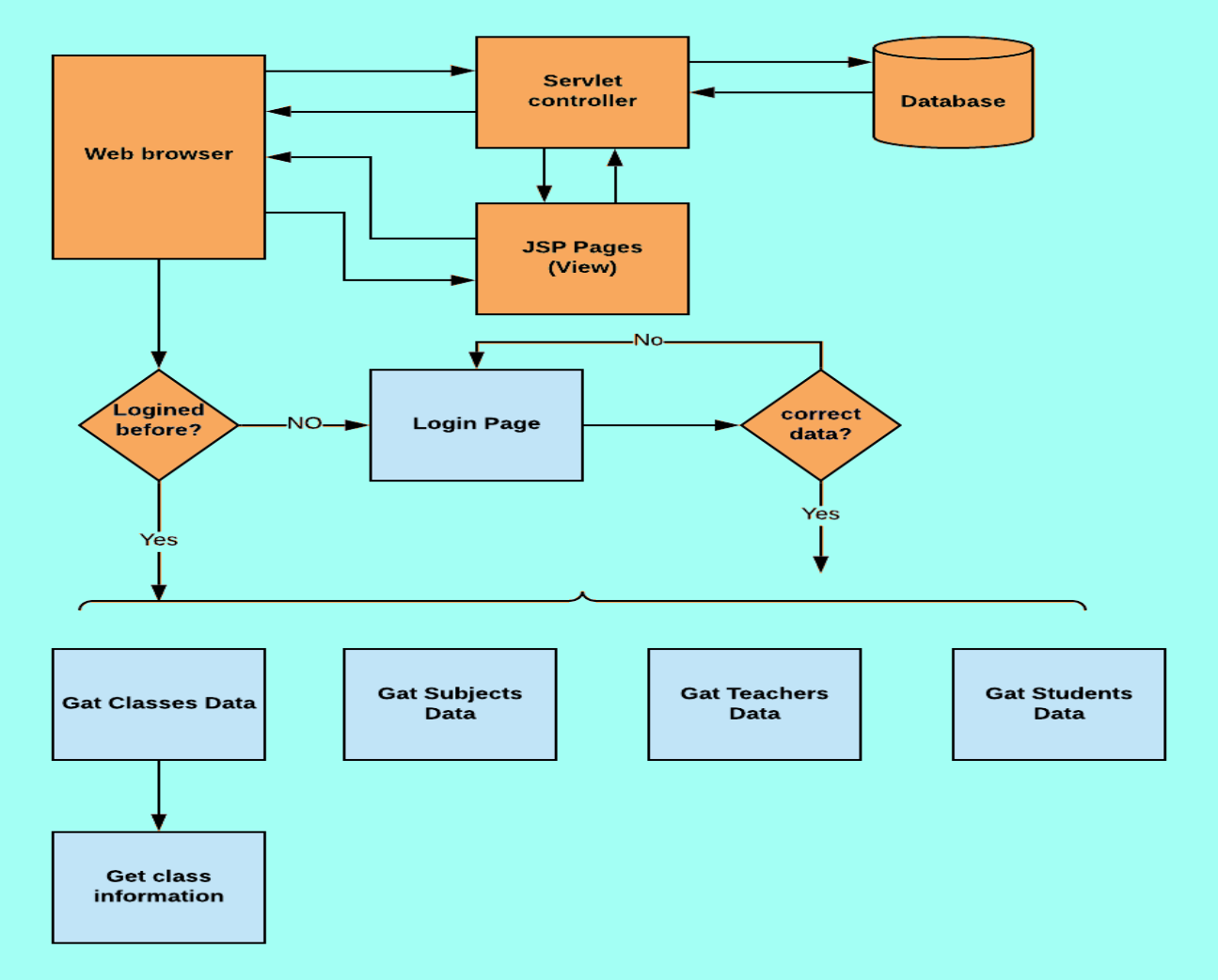
Sprints planning

The project is planned and completed in a single sprint.

Tasks completed in Sprint:-

- Creating the flowchart to determine the flow of the program
- Initializing git environment for project establishment
- Writing java code to fulfill the requirements
- Testing and debugging programs with different inputs
- Pushing code to GitHub.
- Creating this specification document highlighting application capabilities, appearance, and user interactions

The flow of the Application



Pushing the code to the GitHub repository

- Initialize repository in project folder using the following command:

  git init

- Add all the files to your git repository using the following command:

  git add .

- Commit the changes using the following command:

  git commit . -m “<commit message>”

- Push the files to the folder you initially created using the following command:

  git push -u origin master

Unique Selling Points of the Application:-

• Object-Oriented: used to create and model objects for users and their credentials.

• Databases: used to store and retrieve data.

• Data Sources: used to define a set of properties required to identify and access the database.

• Collections: used some collections such as arraylist to store collection of data.

• Exception Handling: used to catch problems that arises in the code especially in I/O blocks.

• Cookies: to store log-in data on the client browser.

Conclusions:-

What more can be done:

- Use of advanced CSS and JavaScript for more responsiveness
- Addition of more validation checkpoints
- Encapsulation of sensitive information like web address which contains some credentials