

SOFTWARE DEVELOPMENT PRACTICES

STUDENT RESULT MANAGEMENT SYSTEM

DONE BY, IT-C

S.V.SACHIN

S.RAKESH KUMAR

M.SATHISH

OBJECTIVE:

The student result management system is to prepare command line interface software at college level to get the consolidated results/progress of the students to facilitate the ease of administration in the college.

USER OF THE SYSTEM:

- System administrator
- Student

FUNCTIONAL REQUIREMENTS:

Depending upon the user role he/she will be able to access only the specific modules of the system.

- 1) Login facility for enabling only authorized access to the system.
- 2) User (with role Data Entry operator) will be able to modify /add/delete information about different students that are enrolled for the course in different years.

OUTPUT CONDITION:

It will display the result of the student which was declared by the respective admin.

NON-FUNCTIONAL REQUIREMENTS:

1 .Safety Requirements The database may get crashed at any certain time due to virus or operating system failure. Therefore, it is required to take the database backup

2. Security Requirements We are going to develop a secured database for the university .Depending upon the category of user the access rights are decided. It means if the user is an administrator thenhe can be able to modify the data, delete, append etc. All other users other than staff only have the rights to retrieve the information about database.

3. Hardware Constraints: The system requires a database in order to store persistent data. The database should have backup capabilities.

4.Availability: 99.99% availability.

5.Standard features: The Student Result Management System (SRMS) is a web- based tool that primarily focuses on delivering results to students and instructors. The student checks their separate outcomes using their university-registered recognition ids, as well as their grades and semester percentages.

6.Monitoring:Should be able to monitor via as-in enterprise monitoring tools.

7.Cloud:The Solutions should be made Cloud-ready and should have a minimum impact when moving away to Cloud infrastructure.

8.Browse compatible:All latest browsers.

TECHNOLOGY STACK:

- HTML & CSS
- JAVASCRIPT
- MYSQL

APPLICATIONS ASSUMPTIONS:

1. The login page should be the first page rendered when the application loads.
2. Manual routing should be restricted by using AuthGuard by implementing the canActivate interface.
3. Unless logged into the system, the user cannot navigate to any other pages.
4. Logging out must again redirect to the login page.
5. To navigate to the admin side, you can store a user type as admin in the database with a username and password as admin.
6. Use admin/admin as the username and password to navigate to the admin dashboard.

VALIDATIONS:

1. Basic email validation should be performed.
2. Password validations should be performed.

PROJECT TASK:

API Endpoints:

USER:

Action URL Method Response:

Login /login POST Pass/Fail

Signup /signup POST Pass/Fail

Total pass/result/{register number} POST result of total pass

Total fail/result/{register number} POST result of total fail

ADMIN:

Action URL Method Response:

Result/Admin GET Array of the result

Pass or Fail/Admin GET Array of the result whether the student is pass or fail

Total pass/admin/total Pass{register number} POST Total pass

Total fail/admin/total Fail{register number} POST Total fail

FRONTEND:

CUSTOMER:

1. Auth: The customer can authenticate login and signup credential.
2. Register: The students has options to sign up and see their result by providing their basic details.

Ids:

- Email
- Username
- Password
- Register number
- Login Button

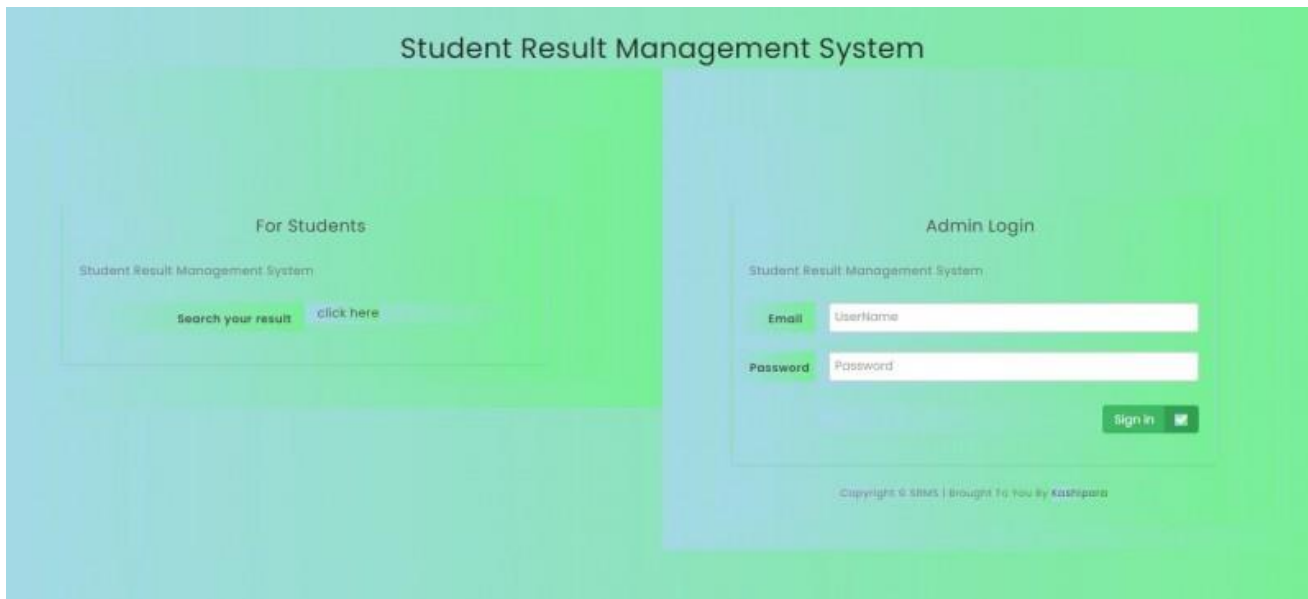
- Register Link
- Login Box

3.Login: Already registered students can log in using the registered email id and password.

Ids:

- Email
- Password
- Register number
- Login Button
- Register Link
- Login Box

LOGIN PAGE:



The screenshot displays the login interface for the 'Student Result Management System'. The page has a light green background. At the top center, the title 'Student Result Management System' is displayed. Below this, there are two main sections: 'For Students' on the left and 'Admin Login' on the right. The 'For Students' section includes a 'Search your result' button and a 'click here' link. The 'Admin Login' section features input fields for 'Email' (labeled 'Username') and 'Password', followed by a 'Sign in' button. At the bottom of the page, a small copyright notice reads 'Copyright © 2018 | Brought To You By Kashipara'.

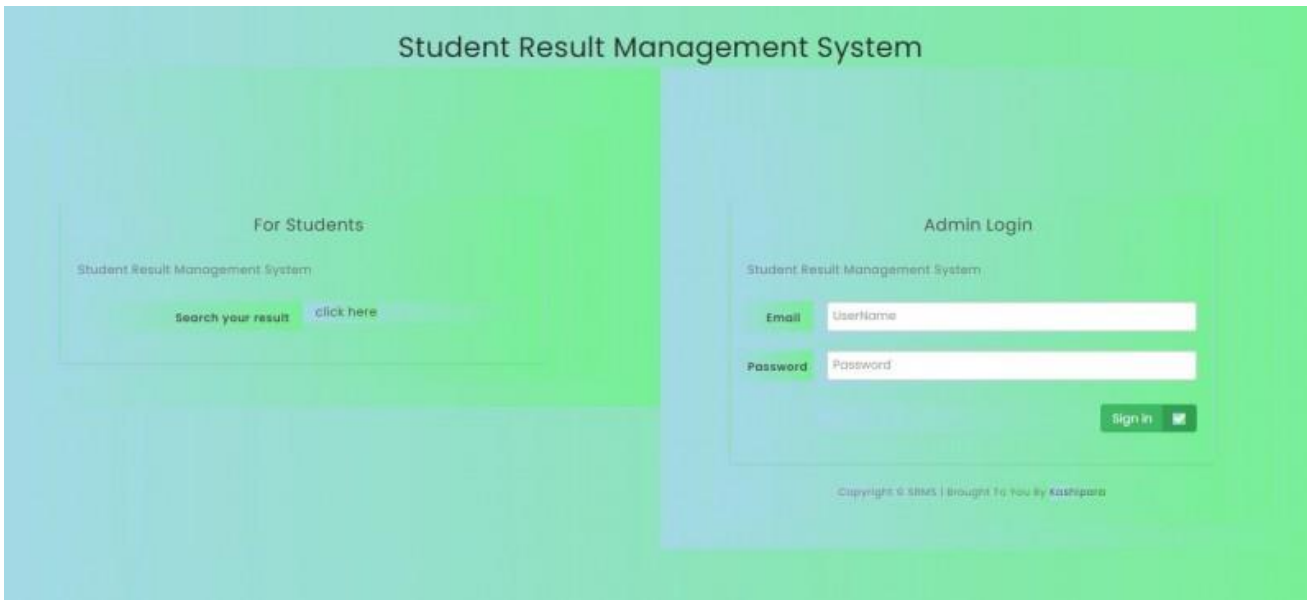
4.Dashboard: A homepage which contains all the details regarding the results (Pass or Fail) of the students.

Ids:

- User Navbar
- Result

- Pass or Fail
- Logout Button

OUTPUT SCREENSHOT:



ADMIN:

Admin dashboard:

A dashboard where the list of the students result is displayed on the admin side.

BACKEND:

Model Layer:

1. User Model: The user type (admin or student) and all user information are stored

a. Attributes:

- Email: String
- Password: String
- Username: String
- Register number: String

- Active: Boolean
- Role: String

2.Login Model: This class contains the email and password of the user.

- Email: String
- Password: String

3.Result Model: This contains the result of the students.

a. Attributes:

- Total Pass: String
- Total Fail: String

CONTROLLER LAYER:

1.Signup Controller: This class control the user signup

a. Methods:

i. Save User(User Model user): This method helps to store users in the database and return Pass or Fail based on the database transaction .

2. Login Controller: This class controls the user login.

a. Methods:

i. CheckUser(Login Model data): This method helps the user to sign up for the application and must return Pass or Fail.

