A

PROJECT PHASE 1

REPORT

On

# Students Skills Evaluation System

Submitted in partial fulfillment of the requirements for the degree of

**Bachelor of Technology** in **Information Technology**

By


1) **Danish Ahemad Shakil Khan (**T2054491246011**)**
2) **Yashraj Vijay Ahirrao (**T20544912460**)**
3) **Tushar Ramkrushna Patil (**T20544912460**)**
4) **Sagar Ajay Dalvi** (T20544912460)


Under the guidance

of

**Prof. Rubi Mandal**


**DEPARTMENT OF INFORMATION TECHNOLOGY**

SHRI VILE PARLE KELAVANI MANDAL'S

**INSTITUTE OF TECHNOLOGY, DHULE**

Survey No. 499, Plot No. 02, Behind Gurudwara, Mumbai-Agra National Highway,
Dhule-424001, Maharashtra, India.

Office Phone: 02562-297801 / 297601, Fax : 02562-297801, Mail: IOTDhule@svkm.ac.in


**Academic Year 2023 – 24**

SHRI VILE PARLE KELAVANI MANDAL'S

# INSTITUTE OF TECHNOLOGY, DHULE

Survey No. 499, Plot No. 02, Behind Gurudwara, Mumbai-Agra National Highway,
Dhule-424001, Maharashtra, India.
Office Phone: 02562-297801 / 297601, Fax : 02562-297801, Mail: IOTDhule@svkm.ac.in

**Academic Year 2023 – 24**

## CERTIFICATE

This is to certify that Mr. Danish Ahemad Shakil Khan

Mr. Yashraj Vijay Ahirrao

Mr. Tushar Ramkrushna Patil

Mr. Sagar Ajay Dalvi

students of Information Technology, bearing has successfully completed project phase-1 report on

**Student's Skills Evaluation System** 29/11/2023 to my satisfaction and submitted the same during the academic year 2023-2024 towards the partial fulfillment of Bachelor of Technology under Dr. Babasaheb Ambedkar Technological University, Leonere, under the guidance of Dr. Bhushan Chaudhari.

| Prof. Rubi Mandal | Dr. Bhushan Chaudhari | Dr. Nilesh Salunke |
|---|---|---|
| **Project Guide & Coordinator** | **HOD** | **Principal** |

Name and Sign with Date
Examiner 1

Name and Sign with Date
Examiner 2

# DECLARATION

We declare that this written submission represents ideas in our own words and where other's ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Signatures

Mr. Danish Ahemad Shakil Khan
Mr. Yashraj Vijay Ahirro
Mr. Tushar Ramkrushna Patil
Mr. Sagar Ajay Dalvi

# Acknowledgments

We have immense pleasure in expressing our interest and deepest sense of gratitude towards our project guide Dr. Bhushan Chaudhari for the assistance, valuable guidance and co-operation in carrying out this project successfully. It is a privilege for us to have been associated with Dr. Bhushan Chaudhari, our Project Guide, during our Project Phase 1 work. We have greatly benefited from his/her valuable suggestion and ideas. It is with great pleasure that we express our deep sense of gratitude to him/her for his/her valuable guidance, ctant encouragement and patience throughout this work. We express our gratitude and are thankful to all people who have contributed in their way in making this final year project phase 1 success. Particularly we want to thank Prof. Rubi Mandal, Project Coordinator for our department for making this process seamless for us and arranging everything so perfectly. I take this opportunity to thank all the class mates for their company during the course work and for the useful discussion, I had with them. We take this opportunity to express our heartfelt gratitude towards the Department of InformationTechnology of Shri Vile Parle Kelvani Mandal's Institute of Technology, Dhule and Dr. Nilesh Salunkhe, Principal of Shri Vile Parle Kelvani Mandal's Institute of Technology, Dhule, that gave us an opportunity for the presentation of our project phase 1 in the esteemed organization and for providing the required facilities in completing this project. We are greatly thankful to our parents, friends and other faculty members for their motivation, guidance and help whenever needed.

## Abstract

The stringent 60% academic criteria imposed by numerous companies serve as a bottleneck, limiting opportunities for students with substantial non-academic achievements. This rigid standard fails to recognize practical skills crucial to a candidate's value. This innovative web application aims to redefine the evaluation process for IT students, surpassing traditional academic metrics. The system comprehensively assesses students based on multiple parameters, including academics, certifications, programming skills, projects, and extracurricular activities such as sports. By incorporating these diverse dimensions, the application provides a holistic understanding of each student's capabilities, strengths, and areas for improvement. The user-friendly interface enables students to input relevant data, including certifications, programming skills, projects, and sports involvement. Educators and evaluators can securely access this information through a dashboard to visualize and analyze individual progress. The application utilizes APIs and advanced algorithms to identify and evaluate students based on various parameters. Additionally, students gain access to their individual progress reports, fostering self-awareness and motivation for continuous improvement. The web application seeks to revolutionize the evaluation of IT students, empowering them with a comprehensive understanding of their abilities for success in both academic and real-world scenarios. Ultimately, it aims to redefine the conventional approach to student evaluation in the Information Technology sector. This transformative approach not only bridges the gap between academic assessments and real-world preparedness but also cultivates a dynamic learning environment that adapts to the evolving demands of the IT industry.

# LIST OF ABBREVIATIONS

| | |
|---|---|
| OCR | Optical Character Recognition |
| TBA | Traditional Benchmark Alternatives |
| WAE | Web Application for Evaluation |
| SIFA | Sports Integration for Achievement |
| VSE | Value of Sports in Education |
| IRRS | Implications for the IT Industry and Real-world Skills |
| APE | Academic Prowess Evaluation |
| RAE | Restrictive Academic Environment |
| AMPA | Acknowledge More Parameters Approach |
| MRPR | Motivation for Redefining Parameters Research |
| CCL | Contribute to Creation of Landscape |
| URP | Unjustly Restricted Proficiency |
| FMD | Fostering More Dynamics |
| FBA | Formidable Barrier Analysis |

## LIST OF FIGURES

# INDEX



# Table of Contents

# 1. Introduction

## 1.1. Introduction

In the rapidly evolving realms of education and employment, the conventional gauges of student competence, primarily academic percentages, often prove insufficient in encapsulating the rich tapestry of diverse talents and skills. In response to this challenge, our project emerges as a transformative initiative in the sphere of student evaluation, particularly within the Information Technology (IT) domain. This report delves into the intricacies of our innovative approach, recognizing the limitations of traditional benchmarks and outlining a comprehensive solution that promises to redefine how we assess IT students, The prevailing practice of imposing a stringent 60% academic benchmark creates inadvertent barriers for numerous students who excel in non-academic domains such as certifications, projects, programming skills, and sports. This exclusivity restricts access to valuable job opportunities, hindering the holistic development of individuals and limiting diversity in the job market. Our project sets out to address this disparity by introducing a web application designed to encompass a broader spectrum of parameters in the evaluation process.

At its core, the application seeks to provide a holistic view of a student's capabilities, moving beyond the conventional academic measures. By incorporating a diverse set of parameters, including certifications, projects, programming skills, and sports achievements, the system aims to offer a more accurate and comprehensive representation of a student's potential value to the IT industry. This multidimensional approach not only acknowledges the multifaceted nature of talent but also aligns with the dynamic demands of the contemporary job market, The user-centric design of the application ensures accessibility and ease of use. Students can input relevant information through an intuitive interface, while educators and evaluators access a secure dashboard for a thorough and nuanced evaluation process. The inclusion of advanced technologies such as APIs, sophisticated algorithms, and Optical Character Recognition (OCR) for sports certification verification further enhances the system's capabilities, ensuring a fair and accurate assessment.

Beyond the technical facets, the project carries a vision of empowerment. By prioritizing transparency and fairness, the system aims to level the playing field, providing opportunities for all deserving candidates. The emphasis on student engagement is a key tenet, encouraging individuals to actively participate in their educational journey and take ownership of their development.

In the dynamic landscape of IT education, our research not only aims to redefine the conventional evaluation methods but also recognizes the multifaceted nature of student development. Beyond academic and technical proficiency, we acknowledge the importance of holistic growth, including physical well-being and the cultivation of essential life skills. This acknowledgment leads us to

introduce a Sports parameter in our comprehensive evaluation system, Sports-integration, Integrating physical activities, such as indigenous sports, into educational practises is a crucial aspect of cross-curricular pedagogy. Critical life skills including cooperation, self-initiative, self-direction, self-discipline, teamwork, and responsibility are intended to be fostered by this approach. Our goal is to give pupils a comprehensive education that goes beyond conventional academic limitations by incorporating sports into the curriculum.

Nep's emphasis on sports aligns seamlessly with our approach, as it recognizes that the skills honed through sports activities are not only beneficial in personal life but also highly relevant in organizational settings. In a professional environment, collaboration, self-initiative, discipline, teamwork, and responsibility are qualities highly sought after by employers. The inclusion of the Sports parameter in our evaluation system is a strategic acknowledgment of the value of these skills in the context of an IT student's future workplace.

As we redefine the evaluation process for IT students, the Sports parameter serves as a means to promote a more comprehensive and well-rounded assessment. It goes beyond technical competencies, emphasizing the importance of physical fitness and life skills that are increasingly recognized as essential in the modern professional landscape. Through this innovative approach By educating them for the demands of the IT business as well as a successful and balanced life outside of the classroom, we hope to support the overall development of IT students.

This report not only outlines the technical aspects of the project but also emphasizes its broader implications for the IT industry and the job market as a whole. The ultimate goal is to contribute to a more equitable and dynamic professional landscape—one that recognizes and rewards the diverse talents and potential within the IT sector. Through this innovative approach, we aspire to bridge the gap between traditional academic evaluations and the real-world skills demanded by today's IT industry, creating a pathway for students to thrive and succeed in their chosen careers.

### 1.2. Motivation behind project topic

In the dynamic landscape of professional opportunities, the stringent eligibility criteria imposed by many companies, demanding a consistent academic performance of 60% or above, emerge as formidable barriers for aspiring individuals. This benchmark, ostensibly designed to gauge academic prowess, inadvertently creates a restrictive environment, impeding access to valuable job opportunities and hindering the career progression of numerous talented individuals.

The motivation underlying this research project is deeply rooted in recognizing the inherent inequities ingrained within such a rigid evaluation system. The prevailing emphasis on academic percentages fails to capture the diverse array of skills and talents that students may possess beyond traditional academic metrics. Consequently, the rigid adherence to the 60% benchmark disproportionately disadvantages individuals who exhibit excellence in areas such as certifications, projects, or programming skills.

The consequence of this inflexible approach is the unjust exclusion of potentially exceptional candidates from consideration. These are individuals who, despite falling short of the conventional 60% threshold, may have demonstrated outstanding proficiency in practical skills and talents crucial to the dynamics of the professional sphere. This exclusionary practice not only stifles individual career prospects but also detrimentally impacts the diversity and vibrancy of the job market, Our motivation to delve into this research project is grounded in the commitment to dismantling these barriers and effecting a paradigm shift in candidate evaluation. We seek to redefine the parameters of assessment, moving beyond the myopic focus on academic percentages to acknowledge and appreciate a more holistic spectrum of skills and talents. By doing so, we aim not only to enhance opportunities for deserving individuals but also to contribute to the creation of a job market that values and leverages diverse talents, ultimately fostering a more dynamic and inclusive professional landscape.

### 1.3. Objectives:

- Diversify Evaluation Criteria: Challenge the conventional 60% academic benchmark by introducing a comprehensive evaluation system that considers certifications, projects, programming skills, and optional sports parameters.
- Develop User-Friendly Interface: Create an intuitive web application interface to facilitate seamless data input for students, ensuring accessibility and ease of use.
- Implement Real-Time Verification: Incorporate knowledge verification tests for certifications and OCR technology for sports certifications to enhance the accuracy and immediacy of the evaluation process.
- Empower Students: Provide individual progress reports to empower students with insights into their development, encouraging a sense of ownership and motivation for continuous improvement.
- Enhance Educator Tools: Integrate a secure dashboard for educators and evaluators, leveraging APIs and sophisticated algorithms to enable comprehensive assessment and visualization of each student's progress.

### 1.4. Scope of the project

This project ambitiously redefines the evaluation of IT students, transcending the limitations of conventional methods. Focused on a holistic assessment, it encompasses certifications, projects, programming skills, and optional sports parameters, challenging the rigid 60% academic benchmark. The development of a user-friendly web application facilitates seamless data input, while APIs and algorithms ensure a nuanced evaluation. Real-time verification processes and OCR technology enhance accuracy. Empowering students through progress reports, the project aims to foster a sense of ownership. Database integration ensures secure information storage, contributing to a comprehensive and innovative approach to student evaluation in the dynamic field of Information Technology.

### 1.5. Organization of report

The report begins with an introduction that sets the stage for the project, providing essential background information and highlighting the motivation behind selecting the topic. It delves into the challenges posed by the prevalent 60% academic benchmark, emphasizing the need for a more inclusive evaluation system that considers a broader spectrum of a student's abilities. The proposed solution is introduced as a web application designed to revolutionize how IT students are assessed.

Following the introduction, the literature review section explores existing knowledge relevant to student evaluation methods. This involves a critical examination of traditional metrics and a review of emerging trends, shedding light on the limitations of current approaches and the need for innovative solutions.

Moving into the methodology, the parameter evaluation framework is elucidated. The certification parameter involves students uploading their certification documents, taking knowledge verification tests associated with each certification, and undergoing a real-time verification process. Each certification's successful completion significantly influences the student's overall evaluation, with marks reflecting the practical application of certified skills.

The project parameter involves students uploading their projects on GitHub, subjecting them to evaluation by a tool like RepoQuester. Marks are assigned based on the quality and complexity of the projects, ensuring a comprehensive assessment of a student's project contributions, Programming skills evaluation requires students to attempt tests aligned with their chosen programming skills. Marks are assigned based on individual performance in solving skill-specific challenges, providing a nuanced evaluation of practical coding abilities, The optional sports activity parameter encourages students to upload sports certificates, with extra points awarded based on the certification level. This parameter aligns with the broader educational goal of integrating sports to cultivate qualities like responsibility, citizenship, teamwork, self-discipline, initiative, and collaboration.
The report then details the integration of all parameters into a database, allowing educators and evaluators to access and analyze individual student progress comprehensively.

In conclusion, the report systematically organizes the project's components, from the rationale behind the chosen topic to the detailed implementation of the proposed evaluation system. The organization ensures a clear and coherent presentation of the project's objectives, methodology, and outcomes.

## 2. Literature Survey

The idea and implementation details for this project were contributed by "Sridhar Chimalakonda & Nuthan Munaiah" who wanted to give engineers a tool to assess software project quality metrics on the GitHub platform. They created RepoQuester, which can function locally and on individual repositories without the need for pre-defined datasets, by altering the Reaper programme that was previously in use. The pull requests ratio and releases were two new metrics added to Reaper's existing ones. They evaluated the speed and accuracy of our tool and found that it can compute the quality metrics for every repository in the dataset in less than 10 seconds.[1]

This Short Text Extracts from Charts and Documents A helpful method for extracting brief text from chart images and texts was proposed by "Rominkumar Busa, Shahira K C" The recommended solution not only beats existing approaches by 18%, but it also has good accuracy in character segmentation. To demonstrate our findings, we used two datasets from MJSynth. We discover that little text may be recovered with a reasonably high accuracy when comparing our findings to earlier methods.[2]

An automatic grading system that can evaluate and score a student's programming assignment's source code is also examined in this review, "Automated Assessment Tools for grading of programming Assignments: Sidhidatri Nayak & Reshu Agarwal" With automated assessment technologies, teachers can evaluate students more constructively, support students in learning from their errors, and provide scores and feedback to students more promptly.[3]

In their study "Data collection and analysis of github repositories and users," "Fragkiskos Chatziasimidis" Finding patterns that connect owner characteristics and open-source project features to project performance was the aim of the current study. Approximately 10,000 individuals and LOOK projects had their data scraped from the immensely popular GitHub repository. Before doing an analysis, we preprocessed the data to achieve value discretization. As a proxy for project success, the quantity of downloads has allowed us to produce six association rules for endeavours that succeed. In summary, when all six principles are applied, successful GitHub projects appear to have the following characteristics.[4]

In their article Software Certification verification and validation difficulties in expert systems, "Vermesan Det & Norske Veritas" Certification professionals may find benefit in verification and validation (V&V) of knowledge-based systems (KBS) just as much as software

developers may. Certification is arguably the most advanced use of V&V information, methodologies, and procedures to date. Static testing techniques can be used to evaluate the knowledge base in great

detail; however, a combination of static and dynamic testing techniques is needed to certify the knowledge base component.[5]

The XML-based Fully Automatic Assessment System for IT Skills, "Hadda Cherroun and Slimane Oulad," In this study, they introduce a novel approach for automatically assessing IT abilities in practical contexts. Information extraction from XML documents is made possible by the useful tools that accompany the XML format and its capabilities. They divided our two-step evaluation procedure into two halves. In the first, we generate an intermediate representation by preprocessing both the student documents and the answer model using an XML format. In the second stage, they take the required competencies and extract them from the teacher's official document as routes in the XML document's tree form. These pathways are contrasted with those found in the student document. [6]

Terri L. Warholak & Marion Slack Determining the Impact of Student Projects Employing the Buxton and Hanney Repayment Model , Student projects make up a sizable element of the curriculum at many universities and engineering institutions. These tasks, which are often designed specifically to promote critical thinking and foundational skills, are very beneficial to students. The perspectives of students and professors, the sharing of intellectual output between students and preceptors/faculty, and the substantial time and effort needed to help students complete projects successfully have been the focus of many evaluations of student projects in the literature.[7]

"Bao-An Nguyen & His-Min Chen" Evaluation scheme for code quality that is used to gauge how much a student has contributed to programming projects. This study proposes a code-quality awareness to facilitate collaborative project management in programming. The architecture of the suggested system is based on Git-driven technologies and continuous integration servers. They also employed open-source and free static code analysis tools to offer automated evaluation features. Each project's contribution from a solid programming foundation was evaluated using code quality metrics. After that, the projects were grouped according to the degree of impact they had. About 75% of the initiatives that employed unsupervised learning were recognised by the RF classifier, demonstrating their exceptional success.[8]

The study "Masayuki HIRAYAMA & Tetsuya YAMAMOTO" compares the creation of software test items by highly and lowly skilled engineers.This study looked at how highly talented and lowly skilled engineers created different test items. From the experimental application, we were able to ascertain that there are certain differences in the procedure for producing illegitimate test items. They also propose a novel method of developing test items that reduces the influence of skill on test item

generation. The experimental assessment also taught us that the proposed method can generate almost same test items without depending on the skill of the test engineers.[9]

"Yuki Akahane & Hiroki Kitaya." Programming assignments that are automatically evaluated are designed and evaluated Guideline An online programming assignment for self-governing scoring systems is presented in the current study. The system tests private classes and methods, as are often addressed in a foundational course, using the JUnit test Reflect API. The outcome test compares the output text from the student programmes with the reference programme using regular expressions. Our university's programming course effectively utilised and reviewed the system. More than 70% of students turned in additional work, and more than 90% of students used the automatic grading feature. After a while, some of the students delivered their work once again to achieve perfection.[10]

"Sita Ramakrishnan and Heinz Schmidt." Examining the Needs for Software Engineering Education in a Semiotics Framework. We have discussed the educational topics that should be covered in a SE curriculum and have created a course structure that works for our community. The Monash SE courses hint at the notion that software systems are made up of systems of signs with specific applications and meanings in business settings. In addition to outlining the requirements for our SE courses, we have also shown how the area of semiotics may be utilised as a framework to analyse the content.[11]

# 3. PROBLEM STATEMENT

The prevalent practice of companies setting a rigid 60% academic criterion inadvertently creates barriers for students with marks below this threshold, limiting their job opportunities. This conventional approach often fails to recognize the practical skills and talents of individuals. To address this, our proposed system seeks to redefine the evaluation process. By assessing students based on multiple parameters, including certifications, projects, programming skills, and sports activities, we aim to provide a more comprehensive and equitable evaluation. This shift ensures that students' potential is acknowledged beyond academic scores, promoting a fair and inclusive approach in recognizing their diverse skills and capabilities.

### 3.1. Project Requirement Specification -

**1. Introduction:**

Define the purpose and scope of the project.

Identify key stakeholders and their roles.

**2. Functional Requirements:**

Clearly outline the features and functions the system must have.

Specify user roles and their respective permissions.

**3. Certifications Module:**

**Document Upload:**

Define the process for users to upload certification documents.

Specify accepted file formats and size limitations.

**Knowledge Verification Test:**

Outline the structure of knowledge verification tests for each certification.

Specify pass criteria and scoring system.

**Real-Time Verification:**

Describe how real-time verification of certification knowledge will occur.

Ensure timely feedback to users.

**Weighted Impact:**

Specify the algorithm for assigning weights to different certifications.

Clarify how marks impact the overall evaluation.

**4. Projects Module:**

**GitHub Submission:**

Define the process for users to submit projects on GitHub.

Specify requirements for project documentation.

**Repoquester Tool:**

Outline how the Repoquester tool will evaluate project quality and complexity.

Define scoring criteria.

**5. Programming Skills Module:**

**Test Participation:**

Specify the types of programming challenges for different skills.

Outline the platform for conducting skill-specific tests.

**Performance Evaluation:**

Define the criteria for evaluating performance in programming challenges.

Specify how marks will be assigned.

**6. Sports Activities Module:**

**Certificate Upload:**

Define the process for users to upload sports certificates.

Specify supported certificate levels (college, district, state, national).

**OCR Verification:**

Describe how OCR technology will verify the certification level.

Outline the optional nature of the Sports parameter.

**7. Database Management:**

**Data Storage:**

Specify the database structure for storing user information.

Ensure security measures for data protection.

**8. Real-Time Progress Reporting:**

Define the parameters included in progress reports.

Specify how educators and evaluators can access real-time data.

**9. Final Score Calculation:**

**Algorithmic Evaluation:**

Define the algorithms used to calculate the final score.
Specify how API integration will support the calculation.

**10. Non-Functional Requirements:**

Specify system performance expectations (response time, concurrent users).
Define security measures to protect user data.

**11. Assumptions and Constraints:**

Clearly state any assumptions made during requirement gathering.
Identify constraints that may impact the project.

**12. Dependencies:**

Outline any external dependencies crucial for project implementation.
Identify potential risks associated with dependencies.

This Project Requirement Specification serves as a comprehensive document outlining the necessary elements for the successful development and implementation of the proposed evaluation system.

# 4. System proposed Diagram

## 4.1. System proposed Diagram

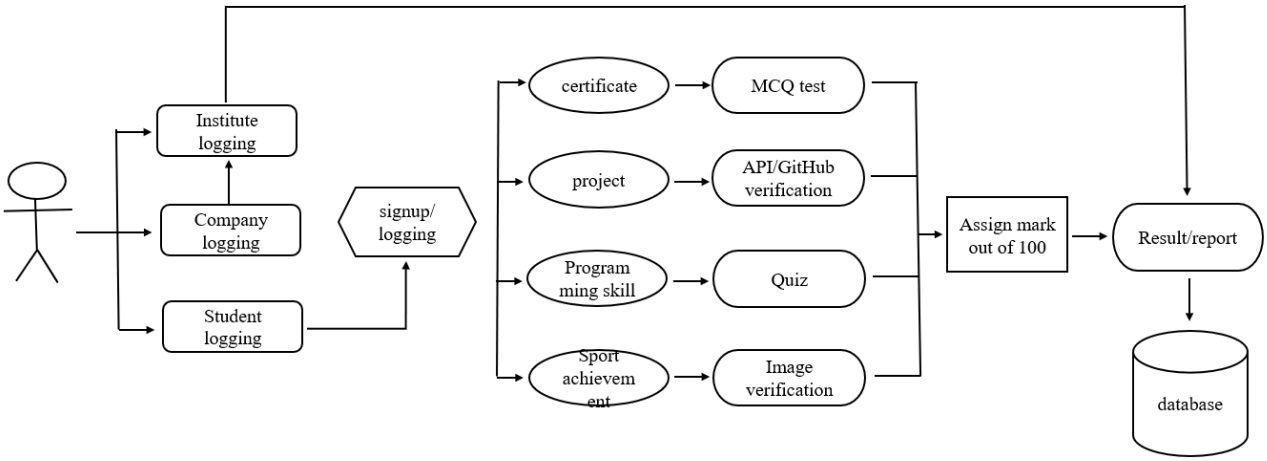


fig 4.1 system architecture

The web application has three main components:

1) Client-side applications: These are the applications that users interact with directly, such as the student app, company app, and institute app. The student app allows students to sign up for quizzes, take quizzes, and view their results. The company app allows companies to create quizzes, assign them to students, and view student results. The institute app allows institutes to verify student certifications.
2) Server: The server is responsible for processing requests from the client-side applications and storing and retrieving data from the database.
3) Database: The database stores all of the data for the web application, such as user profiles, quiz questions and answers, and student results.

The different components of the web application interact with each other as follows:

1. A student signs up for a quiz using the student app. The student app sends a request to the server to create a new user profile and to register the student for the quiz.
2. The server creates a new user profile for the student and registers the student for the quiz in the database. The server then sends a response back to the student app indicating that the student has been successfully registered for the quiz.

3. The student takes the quiz using the student app. The student app sends a request to the server for the quiz questions and answers.
4. The server retrieves the quiz questions and answers from the database and sends them back to the student app.
5. The student answers the quiz questions and submits their answers to the server.
6. The server grades the quiz and stores the student's results in the database. The server then sends a response back to the student app indicating the student's score.
7. A company can view the results of the quiz by logging into the company app. The company app sends a request to the server for the quiz results.
8. The server retrieves the quiz results from the database and sends them back to the company app.
9. An institute can verify a student's certification by logging into the institute app. The institute app sends a request to the server to verify the student's certification.
10. The server verifies the student's certification in the database and sends a response back to the institute app indicating whether or not the certification is valid.

The web application also includes a GitHub integration for verification. This allows users to verify their programming skills by linking their GitHub account to their Quizz account. The Quizz platform can then scan the user's GitHub profile for evidence of their programming skills, such as public repositories, contributions to open source projects, and participation in hackathons.

Overall, the Quizz web application is a complex platform that uses a variety of technologies to provide a comprehensive solution for programming skills assessment and verification.

### 4.2. Proposed Methodology

**1. Parameter Evaluation:**

**Introduction:** The structured framework of Parameter Evaluation assigns specific weightage to each criterion, including Certifications, Projects, Programming Skills, and optional Sports Activities.

**Weightage Allocation:** Each parameter carries a designated weightage contributing to the overall score. The optional Sports parameter provides additional points based on the level of sports certification.

**2. Certifications:**

**Document Upload:** Students upload certification documents, facilitating their inclusion in the evaluation.

**Knowledge Verification Test:** Upon uploading, a knowledge verification test aligned with the certification assesses students' understanding.

**Real-Time Verification:** The real-time verification process allows prompt assessment of a student's certified skills.

**Weighted Impact:** Marks are assigned based on performance in the knowledge verification test, impacting the overall evaluation.

**3. Projects:**

**GitHub Submission:** Project evaluation involves uploading projects to GitHub for assessment.

**Repoquester Tool:** A specialized tool, Repoquester, evaluates the quality and complexity of projects, assigning marks accordingly.

**4. Programming Skills:**

**Test Participation:** Students engage in skill-specific tests, solving challenges to demonstrate practical proficiency.

**Performance Evaluation:** Marks are assigned based on individual performance in solving skill-specific challenges.

**5. Sports Activities:**

**Certificate Upload:** Students upload sports certificates, with extra points assigned based on the certification level.

**OCR Verification:** The optional Sports parameter's certification level is verified using OCR technology.

**6. Database Management:**

**Data Storage:** All information is stored in a secure database, accessible to educators and evaluators.

**Real-Time Progress Reporting:** The database facilitates real-time progress reporting, empowering students with continuous insights.

**7. Final Score Calculation:**

**Algorithmic Evaluation:** APIs and advanced algorithms are leveraged to calculate the final score out of 100.

**Comprehensive Assessment:** The final score offers a holistic representation of a student's capabilities across multiple parameters.

This methodology ensures a systematic and transparent evaluation process, considering diverse aspects of a student's skill set and providing a fair and accurate representation of their potential in the Information Technology field.

# 5. High Level Design of the project

## 5.1. Use case diagram

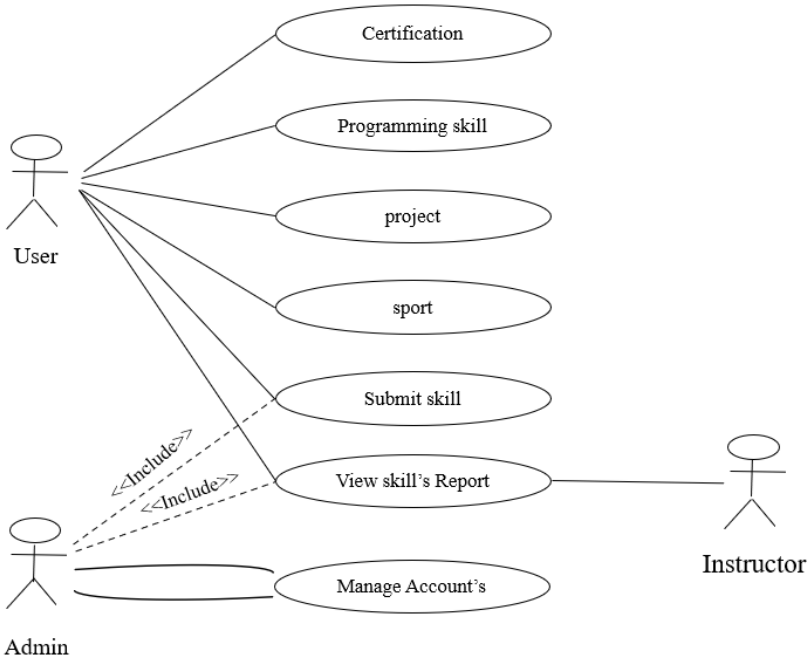


Fig 5.1 use case diagram

**Actors:**

**1. Student:**

Role: The primary individual whose skills are being evaluated within the system.

Interactions: Can view their skills report to track progress and performance.

Submits their skills for evaluation across different categories like programming, projects, sports, etc.

**2.Instructor:**

Role: Likely responsible for assessing or reviewing the students' skills.

Interactions: Accesses the skills reports to evaluate and provide feedback on the student's performance.

**3. Admin:**

Role: Responsible for overseeing and managing the system.

Interactions: Manages user accounts, ensuring the system's integrity and access control.

Includes functions related to viewing skills reports and submitting skills within their administrative duties.

**System:**

**Student's Skills Evaluation System:**

Purpose: This system serves as a platform for evaluating and managing the skills of students.

**Components:**

View Skills Report: Allows users (students and instructors) to access comprehensive reports detailing the student's skills evaluation across various categories.

Submit Skills: Provides a means for students to submit their skills for assessment.

Manage Accounts: A functionality handled by admins to oversee and maintain user accounts within the system.

Certifications: Likely a part of the system where students' certifications or qualifications are managed.

Programming Skills, Projects, Sports: Different categories or aspects of skills that the system evaluates and manages for students.

**Relationships and Interactions:**

Student-System Interaction: Students interact directly with the system components like viewing reports, submitting skills, and engaging with various skill categories.

Instructor-System Interaction: Instructors access the system to review and assess students' skills through the skills reports.

Admin-System Interaction: Admins oversee the system, managing user accounts and ensuring the functionalities related to skills reports and skills submission are functioning properly.

**Overall Functionality:**

The system is designed to facilitate the assessment, tracking, and management of students' skills across multiple domains or categories. It involves students submitting their skills, instructors reviewing these submissions, and admins managing the system and user access.

## 5.2. class diagram

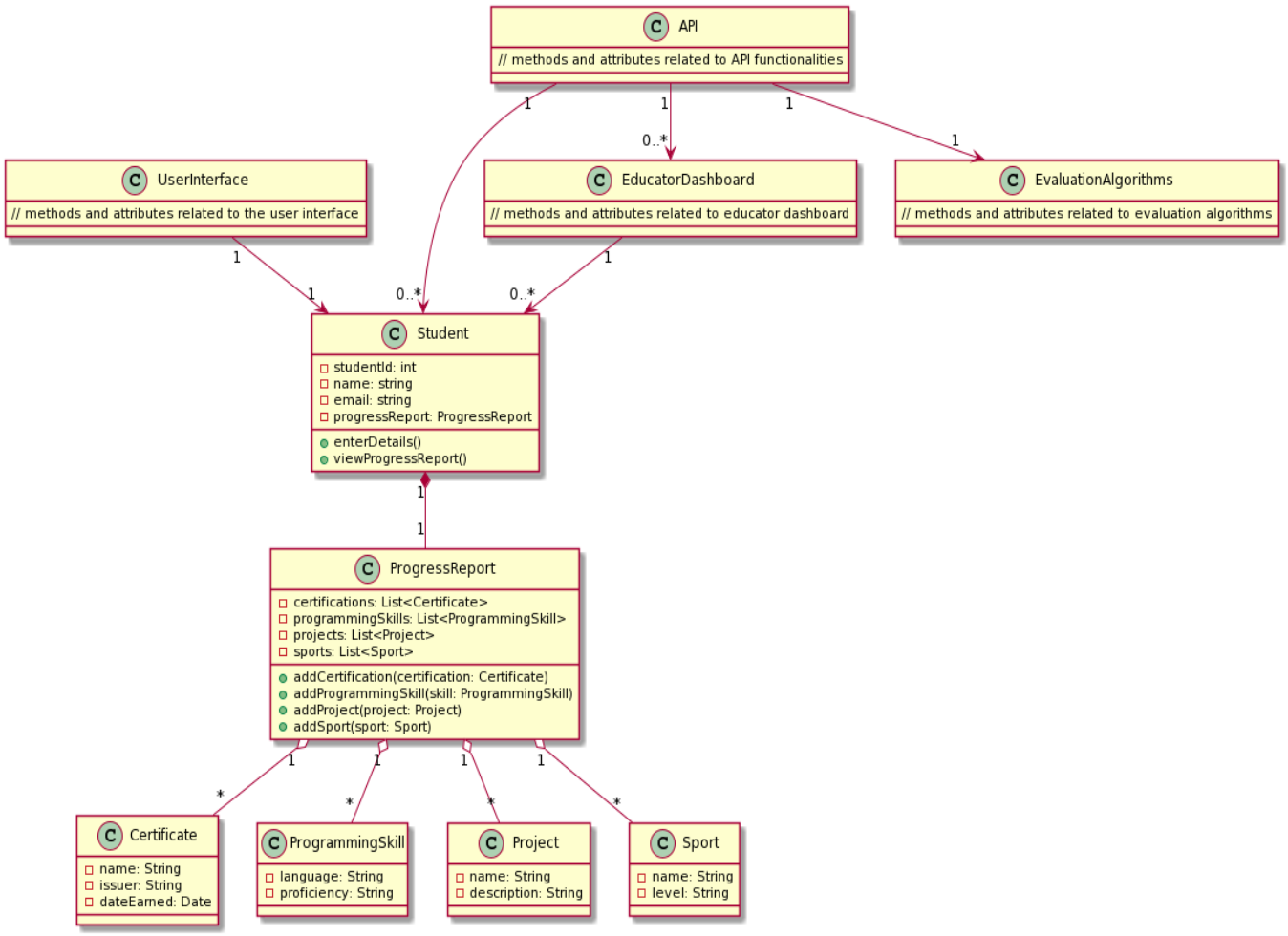


Fig 5.2 Class diagram

This code outlines a set of classes in an object-oriented design, describing a system to manage student information, progress reports, and interactions between different system components.

**Classes:**

1.Student:

Represents a student entity with attributes such as `studentId`, `name`, `email`, and a reference to their `ProgressReport`.

Methods include `enterDetails()` and `viewProgressReport()` for interacting with the system.

**2.ProgressReport:**

Contains information about a student's progress.

Includes lists of `Certificate`, `ProgrammingSkill`, `Project`, and `Sport`.

Provides methods to add items to these lists (`addCertification()`, `addProgrammingSkill()`, `addProject()`, `addSport()`).

**3. Certificate, Programming Skill, Project, Sport:**

Represent specific aspects within a student's progress report, each with its own attributes and encapsulation.

**4. UserInterface:**

Represents the user interface component of the system.

Likely contains methods and attributes for handling user interactions.

**5. EducatorDashboard:**

Represents a dashboard interface for educators or evaluators.

Might include functionalities specific to educators, allowing them to access student information or reports.

**6. API:**

Represents an interface to the system's functionalities, likely for external or internal use.

Associated with `Student`, `EducatorDashboard`, and `EvaluationAlgorithms` for interaction purposes.

**7. EvaluationAlgorithms:**

Presumably contains methods and attributes related to the evaluation or assessment algorithms used within the system.

**Associations:**

**Student-ProgressReport Relationship:**

Each `Student` has one `ProgressReport`.

The `ProgressReport` contains lists of various items related to a student's progress.

**Other Associations:**

`UserInterface` interacts with `Student`.

EducatorDashboard` interacts with `Student` and `API`.

`API` interacts with `Student`, `EducatorDashboard`, and `EvaluationAlgorithms`.

**Overall Functionality:**

- The `Student` class represents individual students within the system, with methods to input details and view progress reports.
- `ProgressReport` encapsulates a student's progress across various aspects like certifications, programming skills, projects, and sports.
- The system seems designed to allow users (students and educators) to interact via a `UserInterface` and `EducatorDashboard`, respectively, with functionalities mediated by an API.
- API serves as a bridge between different components and interacts with `EvaluationAlgorithms`, presumably used for evaluating student progress or generating reports.
- The design appears to facilitate student information management, progress tracking, and interactions between users and the system via interfaces and APIs, providing a structured way to manage and assess student progress.

### 5.3 Object diagram

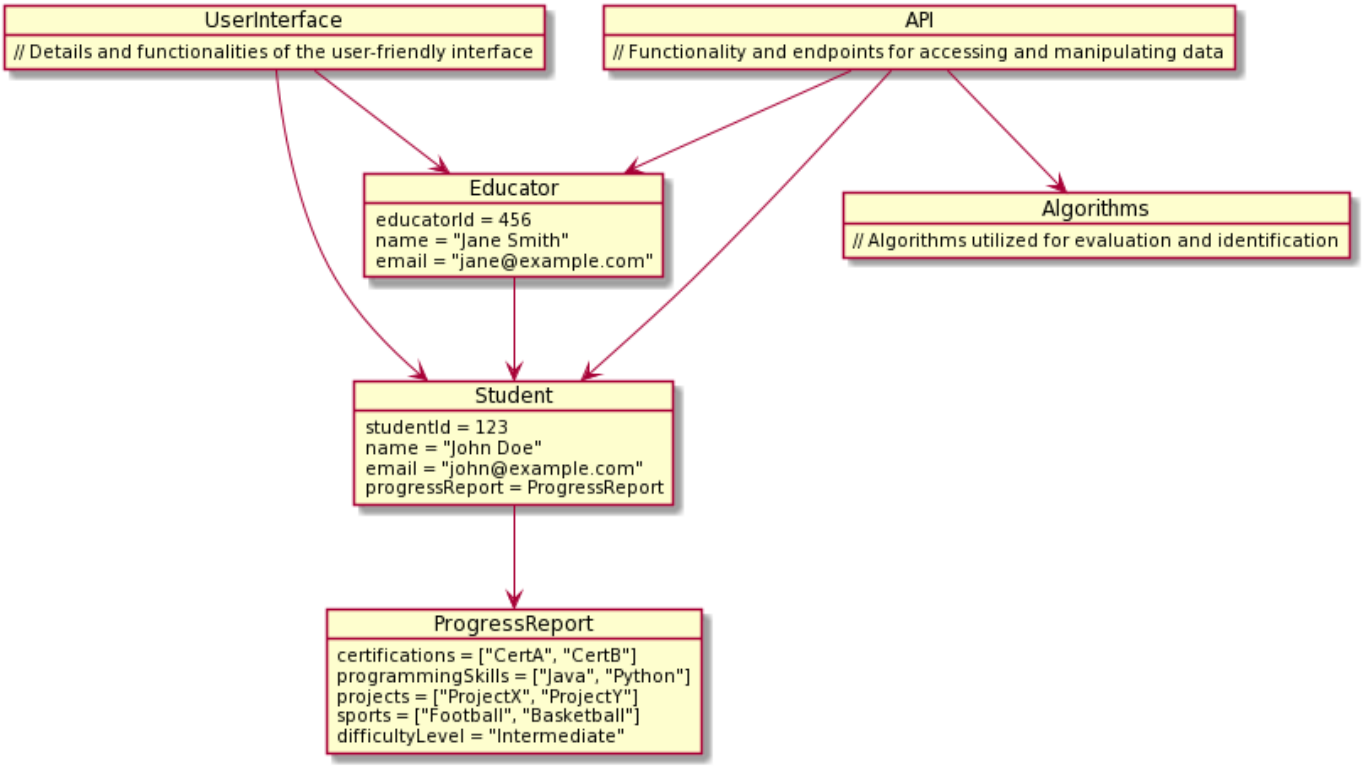


Fig 5.3 Object diagram

This code seems to be a simplified representation using pseudo-code or a language-agnostic structure to demonstrate relationships between different entities in a system, possibly related to an educational context. Let's break it down:

**1.Student Object:**

StudentId, name, email: Basic information about a student.progressReport: This seems to reference the ProgressReport object which contains detailed information about the student's progress, certifications, skills, projects, sports, and difficulty level

**2.ProgressReport Object:**

**Certifications:** Holds a list of certifications achieved by the student (CertA, CertB).

**ProgrammingSkills**: Indicates the programming languages the student is proficient in (Java, Python).

**Projects:** Lists specific projects the student might have worked on (ProjectX, ProjectY).

**Sports:** Lists sports activities the student might be involved in (Football, Basketball).

**DifficultyLevel:** Represents the student's level of proficiency, marked as "Intermediate" in this case.

**3.Educator Object:**

EducatorId, name, email: Basic information about an educator or teacher.

**4. UserInterface Object:**

This entity seems to represent the user interface of a system, possibly providing a user-friendly interface to interact with student and educator data.

**API Object:**

Represents an Application Programming Interface, likely responsible for managing and providing access to student and educator data. It interacts with both Student and Educator objects and also has a connection with Algorithms.

**Algorithms Object:**

Possibly contains algorithms used within the system for evaluation, identification, or other purposes. The API interacts with this object, suggesting that the API might use these algorithms for some functionalities.

The arrows (-->) indicate relationships or connections between these entities.

For example:

The Student object holds a reference to a ProgressReport, indicating that a student is associated with a progress report.

Educator is associated with Student, suggesting a link between educators and the students they teach.

UserInterface interacts with both Student and Educator, suggesting that it provides a way to view or manipulate their information.

API interacts with Student, Educator, and Algorithms, implying it serves as a bridge between the system entities, allowing access to data and functionalities related to students, educators, and algorithms.

This code structure seems to demonstrate the relationships and interactions between various components within an educational system, likely portraying how data flows or is accessed within the system.

### 5.4. Sequence diagram

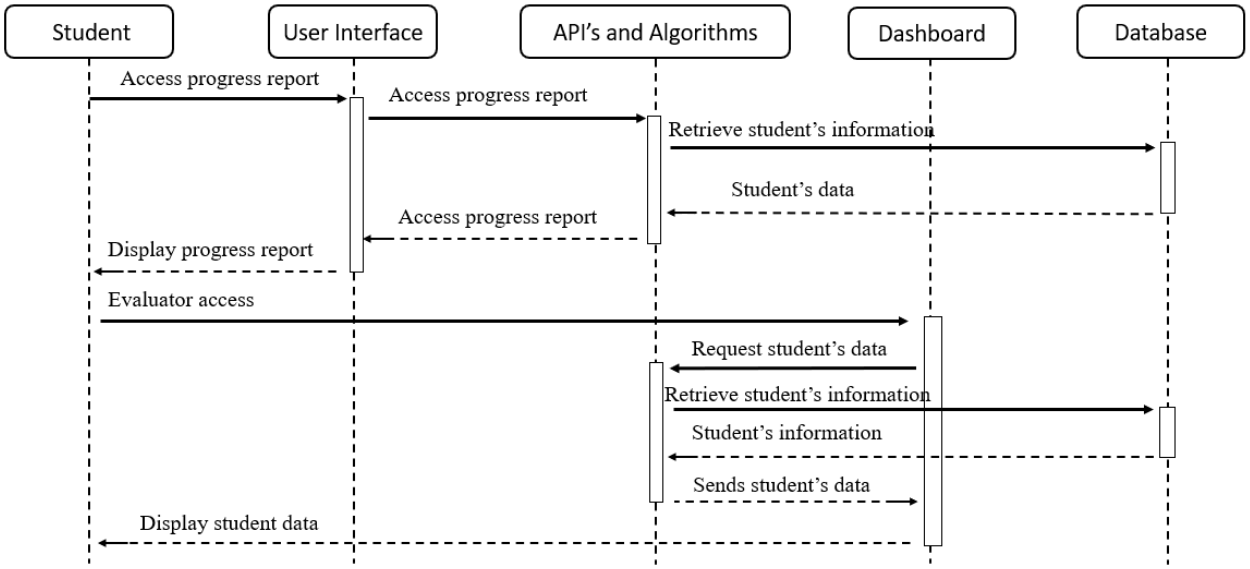


fig 5.4 sequence diagram

**Participants:**

**1.Student:**

Role: Initiates the process by accessing their progress report.

Actions: Interacts with the User Interface to request and receive their progress report data.

**2. User Interface (UI):**

Role: Acts as the interface through which the student interacts with the system.

Actions:

Receives Request: Accepts the student's request for the progress report.

Communicates with Backend: Sends requests to the Backend (APIs and Algorithms) to fetch the student's data.

Receives and Displays Data: Shows the progress report data to the student once received from the Backend.

**3.APIs and Algorithms (Backend):**

Role: Handles processing and communication between the UI and the Database.

Actions:

Receives Requests: Accepts requests for student data from the UI.

Communicates with Database: Engages in communication with the database to obtain the necessary student data.

Sends Data to UI: Provides the student's data back to the user interface for presentation.

**4. Dashboard:**

Role: Represents an educator or evaluator accessing the student's data.

Actions:

Requests Student Data: Initiates a request for the student's information.

Engages with Backend: Sends queries to the Backend to obtain the student's data. Gathers and

Presents Data: The Dashboard screen shows the student's data.

**5. Database:**

Role: Stores and manages the student data.

Actions:

Receives Requests: Accepts requests for student information from the Backend.

Sends Data: Provides the desired student data in response to the backend.

**Sequence of Interactions (Expanded):**

Student Accesses Progress Report: The student interacts with the UI to access their progress report

UI Interaction with Backend: UI forwards the request to the Backend for the student's data.

Requests from the Back End Data from Database: In order to obtain the student's information, the Backend initiates and talks with the Database.

Student Data is Sent by Database: The Database provides the requested student data to the Backend.

Backend Sends Data to UI: The Backend forwards the received data to the UI.

UI Displays Progress Report to Student: The UI receives the data and displays the progress report to the student.

Educator/Evaluator Accesses Dashboard: An educator or evaluator uses the Dashboard to access the student's data.

Dashboard Sends Request to Backend for Data: Dashboard makes a request for the student's data to the Backend.

Backend Retrieves Data from Database: In order to retrieve the student's data, the Backend makes contact with the Database.

Database Sends Student Data to Backend: In response, the database provides the Backend with the student's data.

Backend Sends Data to Dashboard: The Backend sends the received data to the Dashboard.

Dashboard Displays Student's Data: The Dashboard interface receives the data and displays it for the educator or evaluator.

**Overall Interaction Flow:**

The sequence diagram outlines a structured interaction flow between the Student, User Interface, Backend (APIs and Algorithms), Dashboard, and Database. It shows the step-by-step process of how each participant communicates to fulfill requests for student data or progress reports, providing insights into both the student-facing interface and the educator-facing dashboard in accessing and displaying this information’s required information.

## 5.5. State diagram

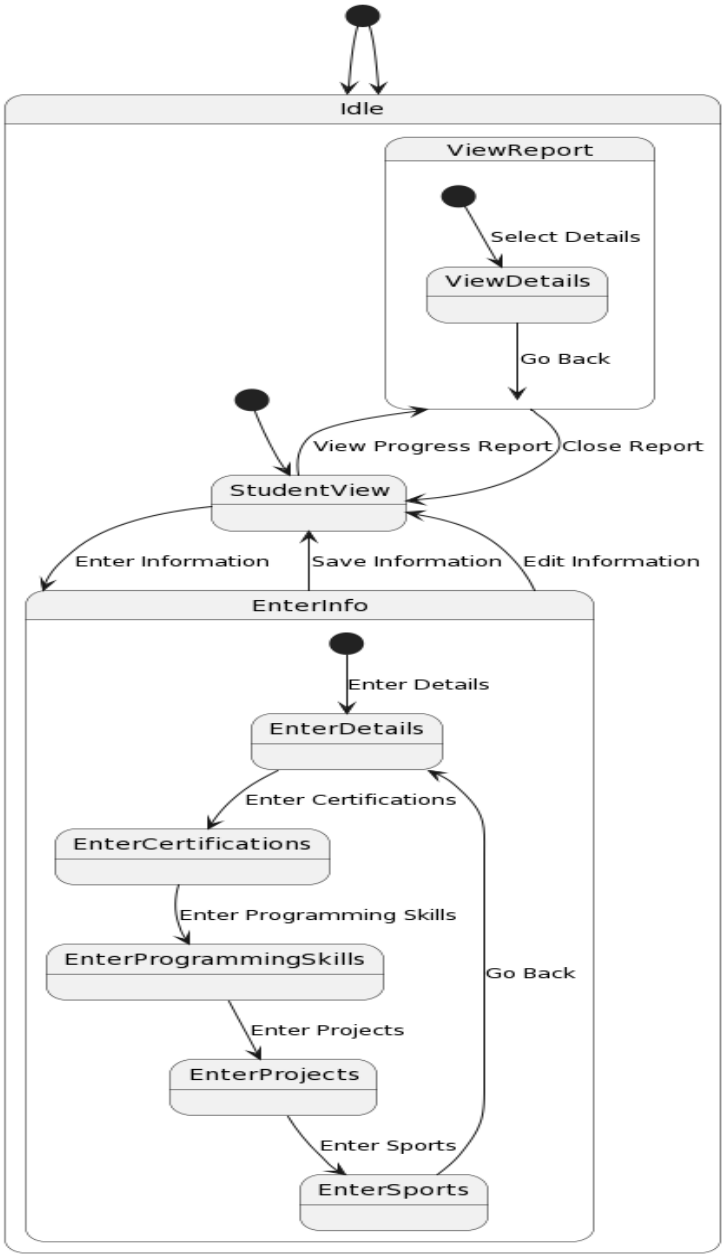


Fig 5.5 state diagram

**States and Transitions:**

**1. Initial State: Idle**

The system starts in an idle state awaiting further actions.

**2. State: StudentView**

Upon initialization, the system moves to the `StudentView` state.

From here, the system allows the student to `Enter Information`, which transitions to the `EnterInfo` state.

**3. State: EnterInfo**

This state involves entering specific information, possibly related to the student's profile or details.

The options include `View Progress Report`, which leads to the `ViewReport` state.

**4. State: ViewReport**

In this state, the student can `Select Details` to view more specific aspects of their progress report.

There's an option to go back to the previous state (`ViewDetails --> ViewReport : Go Back`).

**5. State: EnterDetails**

This state seems to encompass entering more detailed information beyond the general profile.

**6. State: EnterCertifications**

A state where the student may `Enter Certifications`.

**7. State: EnterProgrammingSkills**

A state where the student might `Enter Programming Skills`.

**8. State: EnterProjects**

Allows the student to `Enter Projects`.

**9. State: EnterSports**

This state involves the option to `Enter Sports`.

There's also a transition to go back to the `EnterDetails` state (`EnterSports --> EnterDetails : Go Back`).

**State Transitions:**

The system's passage from one state to another is represented by the transitions (arrows).

The flow from the starting state ({[*]{) to the {Idle} state is depicted in the diagram, and it then moves to different stages in response to user input.

Arrows indicate the direction of transitions between states, showing the progression through different stages of information entry and report viewing.

**Overall Functionality:**

This diagram likely represents a system interface for students to interact with their progress report and enter various types of information, including personal details, certifications, programming skills, projects, and sports. The flow allows for navigation between different states or sections within the system, enabling students to access specific information or input additional details as needed.

Each state represents a stage in the interaction, guiding the user through different functionalities or options available within the system. The transitions indicate how a user can navigate through these stages, providing a structured pathway for interacting with and entering information into the system.

# 6. FEASIBILITY STUDY

## 6.1. Introduction to Feasibility Study :

In the dynamic landscape of today's job market, the traditional approach of evaluating candidates solely based on academic achievements has come under scrutiny. Many companies have instituted a stringent eligibility criterion, requiring a minimum of 60% throughout academics. While such benchmarks may be intended to filter out candidates and streamline the hiring process, they inadvertently pose significant barriers for numerous students, potentially limiting their access to valuable job opportunities.

The goal of this feasibility study is to assess the 60% academic benchmark's impacts on inclusion and diversity of job seekers. By delving into the rationale behind this criterion and its consequences, we seek to determine the feasibility of alternative evaluation methods that better reflect a candidate's potential value to an organization. The study will explore whether a more holistic approach, considering factors such as certifications, projects, and programming skills, can give a more realistic picture of a candidate's potential contributions and skill set.

The justification for this feasibility study arises from the recognition that academic performance, while undoubtedly important, may not be the sole indicator of an individual's competency and potential success in a professional setting. Many students, despite excelling in practical aspects such as hands-on projects and acquiring relevant certifications, find themselves unjustly excluded from job opportunities due to falling short of the rigid 60% benchmark. This exclusion not only affects individual career prospects but also contributes to a lack of diversity in the workforce, as talented individuals from various backgrounds may be overlooked.
As we navigate through this feasibility study, we will analyze the prevailing industry norms, engage with stakeholders in the recruitment process, and evaluate success stories from organizations that have adopted alternative evaluation methods.

The ultimate objective of this feasibility study is to provide a thorough understanding of how the 60% academic standard affects the labor market and to suggest workable alternatives that give candidates' abilities and practical skills priority. By doing so, we aim to contribute to a more inclusive and diverse employment landscape that recognizes and values the diverse strengths individuals bring to the professional realm beyond their academic transcrip

### 6.2. Economic Feasibility:

A key component of our study is its economic viability, which aims to evaluate the cost-effectiveness of using different evaluation techniques during the hiring process, especially when compared to the current 60% academic criterion. This analysis will involve a comprehensive examination of the costs and benefits associated with adopting a more holistic approach to candidate evaluation.

#### 1. Cost Analysis:

**Implementation Costs:** Evaluate the initial expenses associated with transitioning to a more inclusive evaluation system. This may include updating recruitment processes, training staff, and implementing new technology if necessary.

**Operational Costs:** Assess the ongoing costs related to maintaining and adapting the new evaluation methods. This involves considerations such as additional training, software updates, and potential changes in the recruitment workflow.

#### 2. Benefit Analysis:

**Diversity and Inclusion Impact**: Analyze the potential benefits of fostering a more diverse workforce. Research indicates that diverse teams contribute to innovation and improved problem-solving, which can positively impact the overall performance and success of an organization.

**Employee Performance and Retention:** Examine whether candidates selected through alternative evaluation methods demonstrate improved job performance and higher job satisfaction, leading to increased employee retention. This analysis could include case studies and industry benchmarks.

**Brand Reputation:** Investigate the potential impact on the company's reputation. Emphasizing inclusivity and valuing practical skills may enhance the organization's image, making it more attractive to a broader talent pool.

#### 3. Return on Investment (ROI) Analysis:

**Short-Term ROI:** Assess the expected time frame for the organization to recoup the initial investment and start realizing positive returns from the adoption of alternative evaluation methods.

**Long-Term ROI:** Evaluate the sustained benefits over an extended period, considering factors such as improved productivity, reduced turnover costs, and enhanced innovation resulting from a more diverse workforce.

**4. Risk Analysis:**

**Implementation Risks:** Identify potential risks associated with the transition, such as resistance from existing staff, technological challenges, or initial disruptions in the recruitment process.

**Market Risks:** Consider external factors that could impact the feasibility of the proposed changes, such as changes in the job market, economic downturns, or shifts in industry trends.

**5. Cost-Benefit Analysis:**

**Qualitative Analysis:** Take into account non-financial elements that might support the overall viability of the suggested adjustments, such as strengthened employer branding, social responsibility, and improved organisational culture.

By carrying out this economic feasibility analysis, we hope to give decision-makers a thorough grasp of the financial ramifications of using alternative assessment techniques, ultimately assisting them in making choices that are in line with both financial concerns and the more general objectives of developing a workforce that is more diverse and inclusive.

### 6.3. Technical Feasibility:

The technical feasibility analysis is crucial for evaluating the practicality and compatibility of implementing alternative evaluation methods in the hiring process. This involves assessing the existing technological infrastructure, the feasibility of integrating new technologies, and the overall technical implications of transitioning from the traditional 60% academic benchmark to a more holistic approach.

**1. Technological Infrastructure:**

**Assessment of Current Systems:** Examine the organization's current technological setup, taking into account the applicant tracking system (ATS), HRMS, and other pertinent programmes. Determine whether there may be compatibility problems and whether system updates or replacements are necessary.

**2. Privacy and Data Security:** Adherence to Rules: Verify that any new procedure or technology complies with privacy and data security laws. Examine the effects on handling and storing data, paying particular attention to candidate information that is sensitive.

**3. Integration and Interoperability:**

**Compatibility with Existing Systems:** Assess whether the proposed alternative evaluation methods seamlessly integrate with existing HR systems and tools. Evaluate the potential need for customization or development of new interfaces to ensure smooth data flow.

**4. Training and Skill Development:**

**Staff Training Needs:** Identify the training requirements for HR personnel and other stakeholders involved in the recruitment process. Evaluate the feasibility of providing training programs to ensure effective implementation and use of the new evaluation methods.

**5. Scalability:**

**Capacity to Scale**: Assess whether the proposed changes can accommodate the organization's growth and evolving recruitment needs. Consider the scalability of the new methods, especially if the organization plans to expand its workforce or diversify its hiring practices.

**6. Technology Costs:**

**Investment in New Technologies**: Estimate the costs associated with acquiring or developing new technologies required for implementing alternative evaluation methods. Consider both initial costs and ongoing maintenance expenses.

**7. User Experience:**

**Candidate and Staff Experience:** Evaluate the impact on the user experience for both candidates and internal staff. Consider factors such as the ease of use of new tools, the intuitiveness of interfaces, and potential adjustments to the user journey.

**8. Testing and Piloting:**

**Feasibility Testing:** To determine whether the suggested modifications are feasible, carry out small-scale trials or pilot projects. Determine and fix any problems or obstacles that may come up technically during the testing stage.

**9. Risk Analysis**:

**Technology Risks**: Identify potential risks associated with the technical aspects of the transition, such as system failures, data breaches, or compatibility issues that could disrupt the recruitment process.

**10. Continuous Improvement:**

**Adaptability and Upgradability:** Evaluate the adaptability of the new methods to evolving technologies and industry best practices. Consider the organization's ability to implement updates and improvements to stay competitive in the long term.

In conducting the technical feasibility analysis, the goal is to provide insights into the compatibility, reliability, and adaptability of alternative evaluation methods within the organization's technological landscape. A better informed decision-making process is made possible by this analysis, which aids decision-makers in understanding the technical difficulties and opportunities related to the suggested modifications.

### 6.4. Behavioral Feasibility:

Behavioral feasibility, within the context of our study on transitioning from a 60% academic benchmark to more holistic hiring practices, is concerned with assessing the willingness and acceptance of key stakeholders—both internal and external—to embrace the proposed changes. This analysis recognizes that successful implementation depends not only on technological and economic considerations but also on the human aspects of the organization.

**1. Stakeholder Perception:**

Employee Buy-In: Evaluate the perceptions and attitudes of current employees, especially those involved in the recruitment process. Assess their openness to embracing new evaluation methods and their understanding of the potential benefits of a more holistic approach.

**2. Organizational Culture:**

Alignment with Values: Examine how well the proposed changes align with the organization's values and culture. Assess whether the shift towards a more inclusive evaluation method complements the existing ethos and mission.

**3. Change Management:**

Resistance and Mitigation: Identify potential sources of resistance to the proposed changes and develop strategies for mitigating resistance through effective change management practices. This includes communication plans, training initiatives, and addressing concerns proactively.

**4. Candidate Experience:**

**Perceived Fairness**: Analyze how candidates perceive the fairness of the new evaluation methods. Consider communication strategies to ensure transparency and clarity in conveying the shift towards a more holistic approach.

**5. Leadership Support:**

Top-Down Support: Assess the level of support from organizational leadership for the proposed changes. Strong endorsement from top management is crucial for overcoming resistance and fostering a culture that values practical skills alongside academic achievements.

**6. Communication Strategies**:

Internal Communication: Develop effective communication strategies to convey the rationale behind the changes and the potential benefits. Ensure that employees at all levels understand the objectives and are informed about the details of the new evaluation methods.

**7. Training and Development:**

Skill Enhancement: Consider training programs to enhance the skills of employees involved in the recruitment process. Addressing any skills gaps ensures that the team is well-prepared to execute the new evaluation methods effectively.

**8. Behavioral Risks:**

Identifying Potential Issues: Identify potential behavioral risks associated with the transition, such as cultural resistance, fear of change, or concerns about job security. Develop strategies to address these issues and foster a positive behavioral response.

**9. Continuous Feedback:**

Feedback Mechanisms: Establish feedback mechanisms to continuously assess the behavioral impact of the changes. Regular feedback loops can provide insights into the evolving perceptions and concerns of employees and candidates.

Behavioral feasibility analysis acknowledges the significance of human dynamics in the success of organizational changes. By understanding and addressing the behavioral aspects, decision-makers can create a supportive environment that facilitates a smooth transition towards a more inclusive and holistic hiring process.

### 6.5. Time Feasibility:

Time feasibility is a critical dimension in our examination of transitioning from a 60% academic benchmark to a more comprehensive hiring approach. This aspect focuses on the realistic timeframe within which the organization can implement the proposed changes, considering factors such as deadlines, resource availability, and the potential impact on day-to-day operations.

**1. Implementation Timeline:**

Phased Approach: Develop a realistic and well-structured implementation timeline that outlines the key milestones and phases of the transition. Consider breaking down the process into manageable stages to ensure a smooth and controlled integration.

**2. Resource Allocation:**

Human Resources: Assess the availability of human resources required for the transition, including staff responsible for implementing the changes, training programs, and communication initiatives. Ensure that there is adequate personnel to support the shift without causing disruptions in existing workflows.

**3. Training Period:**

Skill Development Timeframe: Determine the time required for training and upskilling employees involved in the recruitment process. Consider any learning curves associated with new technologies or evaluation methods and allocate sufficient time for comprehensive training programs.

**4. Deadline Alignment:**

Alignment with Organizational Deadlines: Ensure that the implementation timeline aligns with broader organizational deadlines and goals. This includes considering any external factors, industry events, or regulatory changes that might impact the schedule.

Time feasibility analysis is instrumental in setting realistic expectations for the implementation of the proposed changes. By carefully considering the time required for various components of the transition and aligning them with organizational objectives, decision-makers can ensure that the process unfolds smoothly and within acceptable timelines.

### 6.6. Resource Feasibility:

Resource feasibility is a vital component of our examination of transitioning from a 60% academic benchmark to a more holistic hiring approach. This dimension assesses the availability and allocation of resources, including financial, human, and technological, required for the successful implementation of alternative evaluation methods.

**1. Accessibility:**

**Resource Allocation for Inclusive Design:** Allocate resources for implementing inclusive design practices in the new evaluation methods. This may involve investing in accessible technologies, training staff on accessibility considerations, and ensuring that the entire recruitment process is accessible to candidates with diverse needs.

**2. Security:**

**Investing in Robust Data Security Measures:** Set aside funds for effective data security procedures. Throughout the hiring process, this entails making investments in encryption technology, conducting frequent security audits, and providing employee training to guarantee the integrity and confidentiality of candidate data.

**3. Scalability:**

Investment in Scalable Technologies: Allocate resources for scalable technologies that can accommodate the organization's growth. This involves investing in systems that can handle increased data volume, user interactions, and additional functionalities as the organization expands.

**4. Efficiency:**

**Investment in Automation and Streamlining Processes:** Allocate resources for the automation of processes to enhance efficiency. This may involve investing in AI-driven tools for resume screening, interview scheduling, and candidate communication to streamline the recruitment workflow and optimize resource utilization.

**5. Integration:**

**Resource Allocation for System Integration:** Allocate resources for seamless integration between existing systems and the new evaluation methods. This involves investing in integration tools, APIs, and potentially upgrading or customizing existing systems to ensure a cohesive and interconnected technology landscape.

**6. User Experience:**

Investment in User Interface (UI) and User Experience (UX) Design: Allocate resources for the design and enhancement of user interfaces to ensure a positive user experience. This includes investing in UX/UI experts, conducting usability testing, and continuously refining the user interface based on feedback for optimal user experience.

Incorporating these resource allocations for accessibility, security, scalability, efficiency, integration, and user experience is essential to ensuring the successful implementation of alternative evaluation methods while maintaining a positive and inclusive environment throughout the recruitment process.

# 7. Experimentation and Results

## 7.1. Block by block results of complete experimentation:

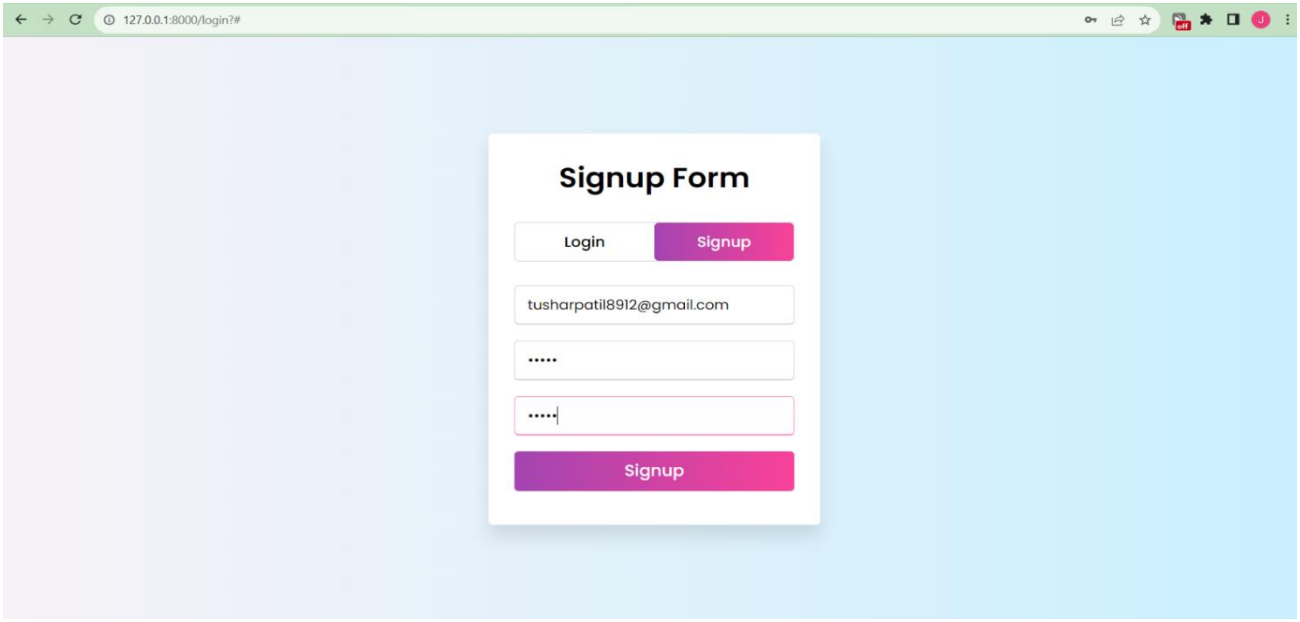


Figure 7.1: Common registration page

**Registration page:**

This is the page where students will enter their credential for registration like email id and password.

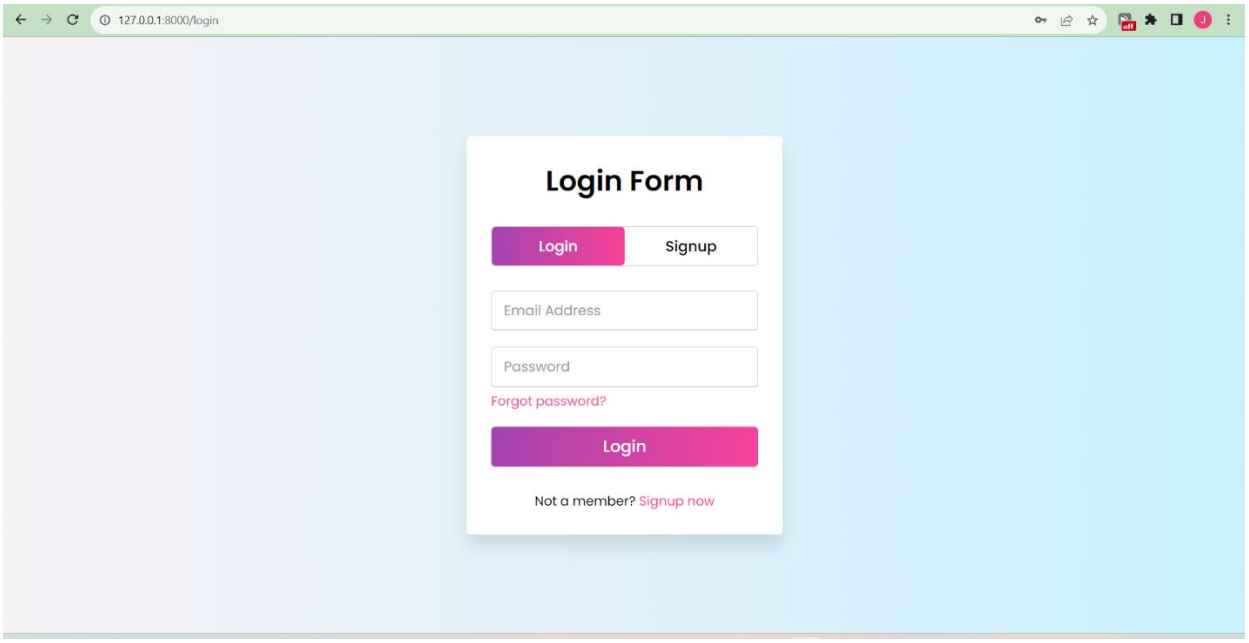


Figure 7.3: Login page

**Login page for students or company:**

This is the login page for students or company or institute where they can access the system by providing credentials which are used at the time of account creation. This allows only the authorized user to login to the system.

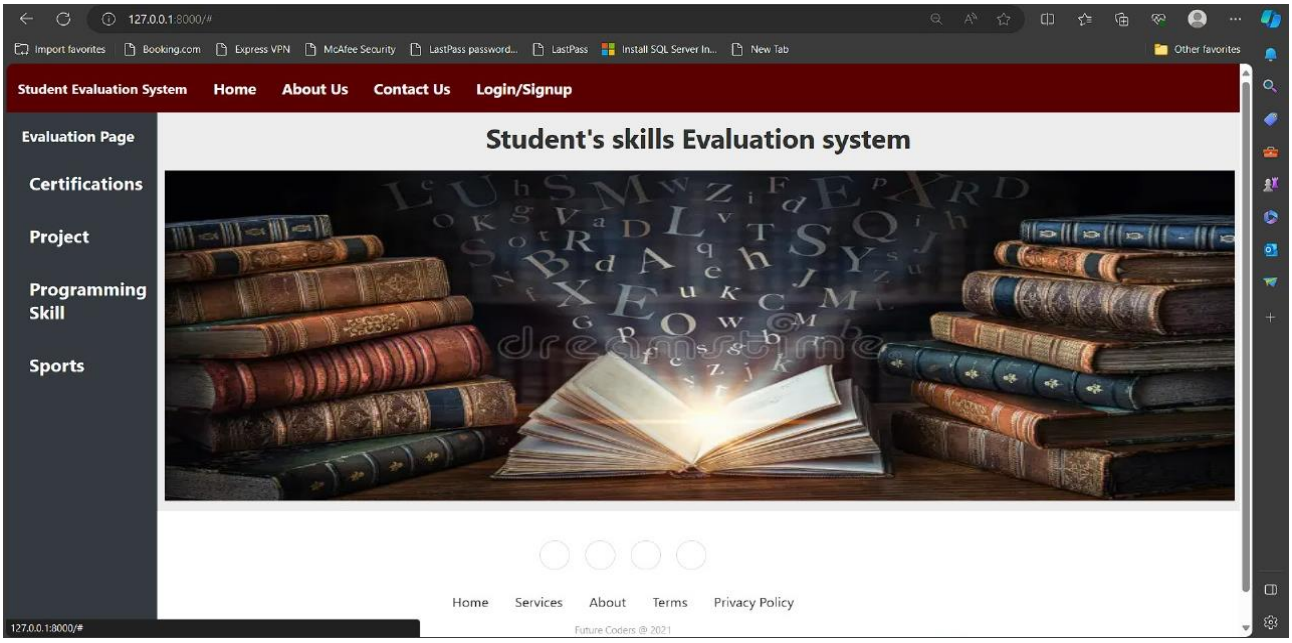


Figure 7.4: Home page for doctors

**Home page for Students:**

This is the home page for students which can be accessible after the successful login. In this page students can give various tests.

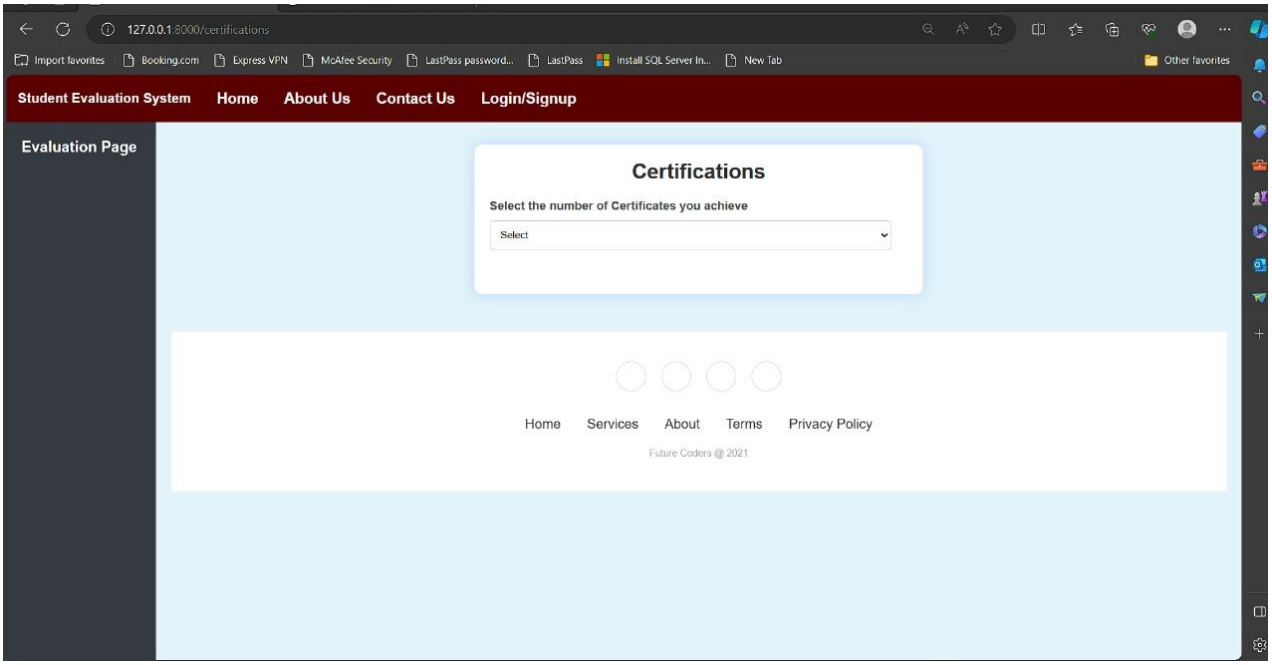


Figure 7.5: Certification page for students

**Certification page for students:**

This is the Certification page for students where they can upload their certificates then will be redirected to test.

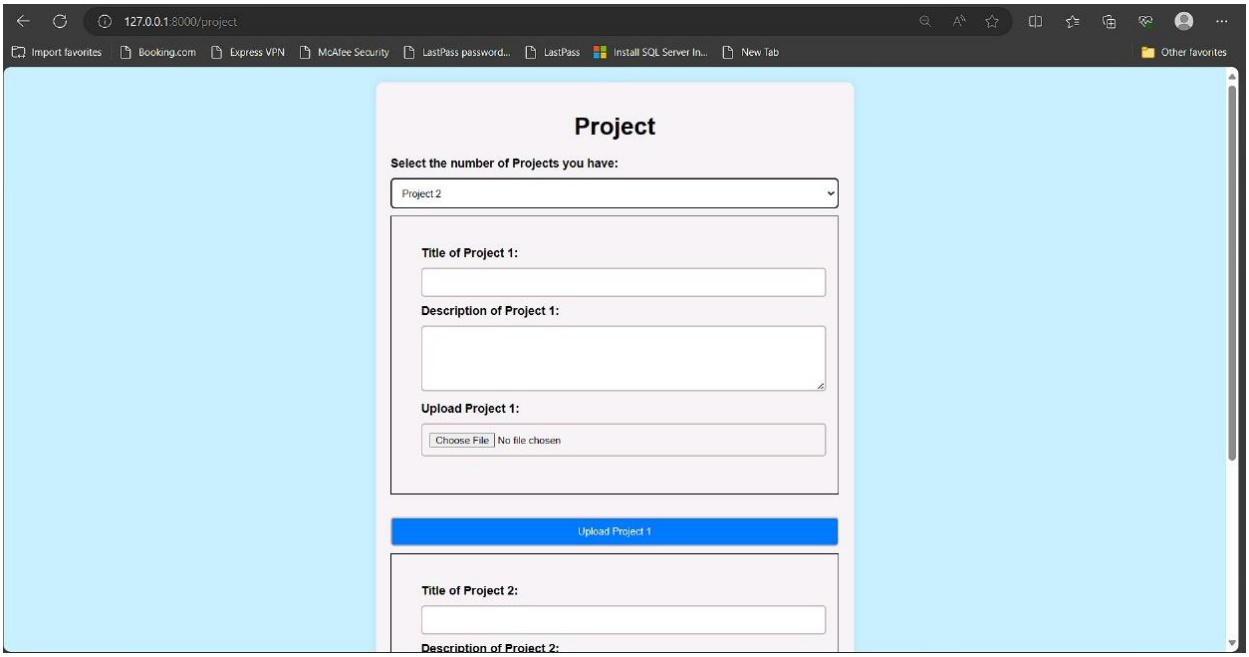


Figure 7.6: project page for students

**project page for students:**

This is the page for project evaluation where students will upload their projects after uploading, the project will be evaluated using RepoQuester tools.

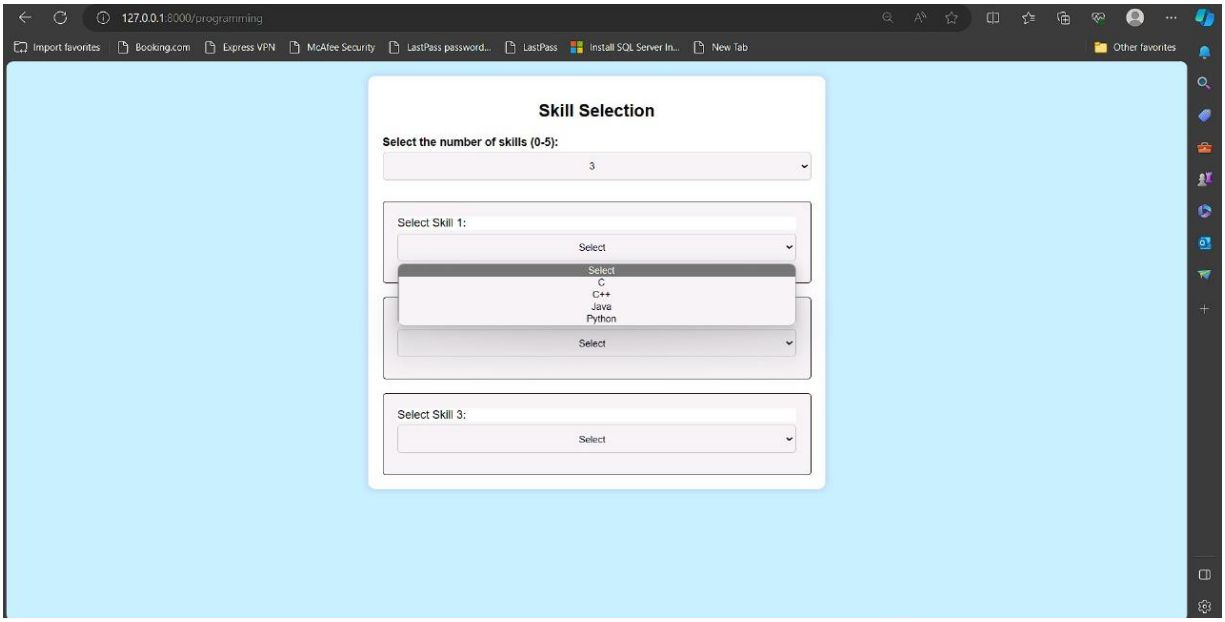


Figure 7.7: programming test page

**programming test page :**

This is the programming test page after students select which language they want to give test they will be redirected to the test**.**

# 7. Conclusion

In conclusion, our research endeavors to address the limitations of the traditional academic evaluation system for IT students. The existing practice of imposing a stringent 60 percent academic benchmark creates barriers for talented individuals, hindering their access to job opportunities and failing to recognize their practical skills. Our proposed web application seeks to revolutionize the evaluation process by considering a diverse set of parameters, including certifications, projects, programming skills, and sports activities. This innovative approach aims to provide a more holistic and equitable assessment, allowing for a nuanced understanding of students' capabilities beyond academic scores. By leveraging APIs and advanced algorithms, our system ensures a fair and detailed evaluation, preparing students for the dynamic IT industry. The inclusion of individual progress reports empowers students to actively engage in their development, fostering a sense of ownership and motivation for continuous improvement. In essence, our research strives to contribute to a more inclusive and dynamic job market that recognizes and values the multifaceted nature of individual potential in the field of Information Technology.

# REFERENCES

[1] RepoQuester: A Tool Towards Evaluating GitHub Projects. Kowndinya Boyalakuntla1, Meiyappan Nagappan2, Sridhar Chimalakonda1, Nuthan Munaiah3 Indian Institute of Technology Tirupati, India

[2] Small Text Extraction from Documents and Chart Images Rominkumar Busa, Shahira K C, Graduate student member, IEEE, and Lijiya A, Member, IEEE.

[3] Automated Assessment Tools for grading of programming Assignments: A review. Sidhidatri Nayak Reshu Agarwal Amity Institute of Information Technology Amity University, India sidhidatri.nayak@gmail.com

[4] Software Certification for Industry – verification and validation issues in expert Systems. Anca I. Vermesan Det Norske Veritas AS Veritasveien 1, 1322 Hovik, Norway Email: Anca.Vermesan@dnv.com

[5] data collection and analysis of github repositories and user. Fragkiskos Chatziasimidis Gnomon Informatics SA 21 Antonis Tritsis Str. 57001 Thessaloniki, Greece +30-2310804150 f.chatziasimidis@gmail.com

[6] XML-based Fully Automatic Assessment System for IT Skills Aicha Chorana, Abdallah Lakhdari, Hadda Cherroun, and Slimane Oulad Naoui Computer science and mathematics department, Faculty of Science

[7] Identifying Student Project Impact Using the Buxton and Hanney Payback Framework. Marion Slack, PhD, Terri L. Warholak, PhD a University of Arizona College of Pharmacy, Tucson, Arizona

[8] Code-quality evaluation scheme for assessment of student contributions to programming projects. 'His-Min Chen & Bao-An Nguyen' Department of Information Engineering and Computer Science, Feng Chia University, Taichung, Taiwan

[9] Analysis of Software Test Item Generation — Comparison between High Skilled and Low Skilled Engineers Masayuki HIRAYAMA , Tetsuya YAMAMOTO TOSHIBA Corporation, Japan

[10] Design and Evaluation of Automated Scoring Programming Assignments Yuki Akahane, Hiroki Kitaya, and Ushio Inoue Graduate School of Engineering Tokyo Denki University Tokyo, Japan

[11] Watson, C.; Li, F.W.B. Failure rates in introductory programming revisited. In Proceedings of the 19th conference on (ITiCSE'14), Uppsala, Sweden, 21–25 June 2014; pp. 39–44.

[12] S.Wang,Y.Han,W.Wu,Z.Hu,"Modelling Student Learning outcomes in studying programming course", Seventh International Conference on Information science and Technology,DaNang,Vietnam,April 16-19,2017.

[13] P. Pickerill, H. J. Jungen, M. Ochodek, M. Mackowiak, and M. Staron, ′ "Phantom: Curating github for engineered software projects using timeseries clustering," Empirical Software Engineering, vol. 25, pp. 2897–2929, 2020.

[14] Amelung, M.; Krieger, K.; Rösner, D. E-assessment as a service. IEEETrans. Learn. Technol. 2011, 4, 162–174.

[15] Mike Joy, Nathan Griffiths, and Russell Boyatt.The boss online submission and assessment system. J. Educ. Resour. Comput. 5, 3, Article2(September2005).