



Sherwood College of Professional Management



AFFILIATED TO,
LUCKNOW UNIVERSITY

PROJECT REPORT ON SKILL MATRIX

(PAPER CODE: BCA 508 P)

SUBMITTED TO
FACULTY, DEPARTMENT OF
COMPUTER APPLICATION

SUBMITTED BY
DEV PRAKASH SINGH
BCA 5TH SEMESTER
2210724040018

Preface

I am glad to have undertaken this project on **Skill Matrix**, which has been a fulfilling learning experience. In this project report, I have included an overview of the project, its objectives, and key activities.

Working on this project has provided me with valuable experience and knowledge, particularly in the area of system analysis and implementation. These insights have significantly enhanced my understanding, and I am excited to share them through this report.

I have made a conscious effort to design the system to be user-oriented and efficient. My goal has been to ensure that users find the system not only practical but also intuitive and helpful.

During the course of this project, I encountered several technical challenges, all of which I successfully addressed with the guidance and support of my teacher, **Mr. Pankaj Mishra**. His expertise and encouragement were instrumental in overcoming these obstacles, and I am deeply grateful for his valuable help and guidance.

I am also thankful to **Sherwood College of Professional Management** for providing the resources and environment necessary to complete this project successfully.

Finally, I extend my heartfelt gratitude to my parents and everyone who supported me during this journey. Their unwavering encouragement and belief in my abilities have been a constant source of motivation.

Thank you,
Dev Prakash Singh
BCA V SEM
2210724040018

Acknowledgment

Fifth Semester Project is a major component of the academic schedule of B.C.A. Hence, I have worked on **Skill Matrix**. The conceptual knowledge acquired by a Management/Computer student is best manifested in the project they undertake.

The present project gives a perfect opportunity to deepen my understanding of the practicalities of information systems across various domains. I express my wholehearted gratitude towards **Sherwood College of Professional Management** for providing me with the opportunity to work on this project and develop my skills in the field of system design and analysis.

I express my heartfelt gratitude to **Mr. Pankaj Mishra** for supervising me during the project period. His guidance, patience, and encouragement have been instrumental in the successful completion of this project. I also extend my special thanks to all the staff members who offered their valuable time and support whenever required.

I am deeply grateful to my parents, who have always been my pillars of strength, providing unwavering support and sound advice at every stage of my academic journey.

Lastly, I wish to acknowledge the contributions of all the respondents, without whom the successful completion of this project would not have been possible. Your input has been vital in shaping this work.

This project has been a remarkable learning experience, and I am truly thankful to everyone who contributed to its success.

Dev Prakash Singh

INDEX

S.No.	Content	Page No.	Signature
1	Chapter 1: Introduction of Skill Matrix	1-11	
2	Overview of the Skill Matrix	1	
3	Background	2	
4	System Objective	2	
5	System Context	2-3	
6	Functional Requirement	4	
7	Non-Functional Requirements	5-6	
8	Future Requirements	6-8	
9	Objectives	8-9	
10	Problem Definition: Skill Matrix System	9	
11	Proposed Solution: Skill Matrix System	9-10	
12	Purpose	10-11	
13	Scope	11	
14	Chapter 2: Problem Formulation	12-19	

15	Introduction	12-13	
16	Feasibility Study	13-15	
17	Problem Specification	15-16	
18	Need for Skill Matrix	16-17	
19	The Proposed System	17	
20	Modules Used in the Project	17-18	
21	Existing System	19	
22	Chapter 3: System Architecture and Design	20-34	
23	Defining a System	20-21	
24	System Development Life Cycle (SDLC)	21	
25	Activates in the SDLC for the Project	21-24	
26	System Analysis	24-25	
27	System Design	25	
28	E-R Diagram	26-27	
29	Data Flow Diagram (DFD)	28-29	
30	Data Tables	30-34	
31	Chapter 4: Testing and Implementation	35-102	

32	Objective of Testing	35-36	
33	Types of Testing	36-39	
34	Recovery and Security	39	
35	Usability Documentation and Procedure	39-40	
36	System Implementation	40	
37	System Specification	40-41	
38	Installation	41	
39	Implementation	41	
40	Screen Shots	42-52	
41	Coding	53-102	
42	Chapter 5: Result and Conclusion	103-105	
43	Result	103	
44	Conclusion	104	
45	Future Scope	104-105	
46	Bibliography	106	

SKILL MATRIX

Chapter 1: Introduction of Skill Matrix

In the rapidly evolving world of information technology, efficient information management plays a crucial role. Over the past few years, computers have gained remarkable popularity, marking the beginning of a technological revolution that has influenced nearly every aspect of human life. Computers, being excellent information processors, are adept at receiving data, performing essential operations, and generating results based on pre-defined programs.

Skill Matrix is a project aimed at providing students with a platform to assess their knowledge level and programming skills. This system acts as a benchmark for students to identify their areas of strength and areas that require improvement.

Overview Of the Skill Matrix

The Skill Matrix project is structured into two primary modules:

1. Admin Module:

The admin module manages and controls all processes within the system. It is responsible for:

- Adding or updating skill assessment criteria.
- Monitoring student activities and progress.
- Generating detailed reports and insights.

2. Student Module:

This module is designed for students to interact with the system. It enables them to:

- Participate in assessments to evaluate their skills.
- Receive feedback and performance analytics.
- Identify areas for improvement in programming and technical skills.

The Skill Matrix project provides a user-friendly platform for both administrators and students, ensuring an efficient skill evaluation process that aligns with the dynamic demands of the IT industry.

SKILL MATRIX

Background

In the traditional approach to skill evaluation, a significant amount of paperwork and human resources were required to conduct any type of skill assessment. By introducing a **Skill Matrix** system, all manual and paper-based work can be eliminated, simplifying the entire process. This system offers a modern solution to test students' skills in a shorter span of time.

One of the significant advantages of this system is the automation of answer evaluation, allowing results to be generated within minutes without manual checking. Furthermore, there are no time constraints on conducting assessments—they can be organized at any time, whether in the morning or late at night.

Additionally, the system provides flexibility to users, enabling them to participate in skill assessments from any part of the world. This innovative approach ensures convenience, efficiency, and accessibility for both students and administrators.

System Objective

In today's era, computers play an integral role in almost every domain, streamlining processes and enhancing efficiency. The increasing dependency on computers stems from their ability to manage and process vast amounts of data seamlessly.

Manually handling large volumes of information is not only cumbersome but also time-consuming and prone to errors. Retrieving specific data quickly, especially during urgent situations, becomes nearly impossible without the use of computers.

The objective of this system is to leverage the power of computers to automate and simplify tasks, ensuring efficient storage, quick retrieval, and streamlined management of information. This not only saves time and resources but also enhances productivity and accuracy in data handling.

System Context

This section outlines the environment, boundaries, and interactions of the Online Examination System (OES) with various entities. It provides a clear understanding of how the system integrates into the existing ecosystem and its role in automating processes.

SKILL MATRIX

The System Context defines:

1. System Environment:

The OES operates within a digital framework, accessible through web-based platforms. It requires a stable internet connection, a server for data management, and user devices such as computers or mobile phones for interaction.

2. System Boundaries:

The system's scope is confined to managing and conducting online examinations. This includes:

- Question paper generation.
- User authentication.
- Answer submission.
- Automated evaluation.
- Results generation.

Tasks outside the system's capabilities, such as the development of question banks or manual interventions during examinations, remain the responsibility of external entities.

3. Interacting Entities:

The system interacts with the following entities:

- Admin: Responsible for creating exams, managing users, and analyzing results.
- Students: End-users who take exams and receive performance feedback.
- Database: Stores questions, user details, results, and other critical information.

4. System Expectations:

- From Admins: Proper configuration of exams and regular maintenance of the system.
- From Students: Compliance with exam rules and timely participation.
- From Supporting Infrastructure: Reliable servers, secure data storage, and uninterrupted internet connectivity.

This delineation helps establish a clear picture of the system's role, its limitations, and its dependencies, ensuring smooth operation and integration into the overall framework.

SKILL MATRIX

Functional Requirement

The functional requirements of the project focus on ensuring that the system operates effectively and meets its objectives. These include:

1. Student Record Management:

- The system should securely store records of all students who have appeared for Skill Matrix assessments.
- These records must include details such as student identification, test scores, and performance analytics.

2. Smooth Examination Process:

- The system should facilitate a seamless Skill Matrix session, ensuring no interruptions during the test.
- Features such as timed sessions, automated question delivery, and answer submission must function reliably.

3. Data Security and Integrity:

- All stored data must be protected against unauthorized access, ensuring privacy and confidentiality.
- Backup mechanisms should be in place to prevent data loss.

4. Automated Evaluation and Results:

- The system should automatically evaluate submitted answers and generate accurate results within minutes.
- Students should be able to view their results immediately after completion.

5. User-Friendly Interface:

- Both admins and students should find the interface intuitive and easy to navigate.
- Clear instructions should guide users through the testing process.

By addressing these functional requirements, the system ensures efficient operation, reliability, and a positive user experience.

SKILL MATRIX

Non-Functional Requirements

Non-functional requirements define the quality attributes and performance benchmarks the system must meet to ensure reliability, efficiency, and user satisfaction. For the **Skill Matrix System**, the key non-functional requirements include:

1. Performance:

- The system should handle multiple concurrent users without noticeable performance degradation.
- Response times for loading pages, accessing records, and submitting answers should not exceed **2 seconds** under normal load conditions.

2. Scalability:

- The system must be scalable to accommodate an increasing number of users or tests without significant changes to the architecture.

3. Availability:

- The system should have a minimum uptime of **99.9%**, ensuring it is available for use at all times, including late-night or early-morning hours.

4. Security:

- Sensitive data such as student records and test results must be encrypted to prevent unauthorized access.
- The system must have secure authentication mechanisms, including strong passwords and session management.

5. Usability:

- The system should have an intuitive and user-friendly interface to minimize the learning curve for both students and administrators.
- It must provide clear instructions and error messages to guide users during interaction.

6. Reliability:

- The system must ensure accurate recording and processing of data without failures or errors.

SKILL MATRIX

- Backup and recovery mechanisms must be in place to restore data in case of unexpected failures.

7. Maintainability:

- The system architecture should allow easy updates and bug fixes.
- Documentation for code and processes should be thorough to support ongoing maintenance.

8. Accessibility:

- The system should be accessible from any location and device, supporting cross-browser compatibility and mobile responsiveness.

9. Trade-offs:

- Performance and scalability may involve trade-offs with cost, and security measures may slightly affect response times. These trade-offs must be carefully evaluated to balance the system's overall functionality and user experience.

By meeting these non-functional requirements, the system will ensure high-quality performance, robustness, and a seamless user experience.

Future Requirements

Future requirements are features and enhancements not currently implemented in the existing version of the **Skill Matrix System** but can be considered for integration in subsequent updates. These improvements may involve advanced technologies or interfaces with other systems to expand the system's functionality and usability.

SKILL MATRIX

Potential Future Enhancements:

1. Integration with AI and Machine Learning:

- Introduce AI-driven analytics to provide personalized feedback to students based on their performance trends.
- Implement adaptive testing, where the difficulty of questions adjusts dynamically based on the student's previous responses.

2. Advanced Reporting and Insights:

- Provide detailed analytics for administrators, including performance comparisons, skill gap identification, and statistical trends.
- Allow students to generate personalized reports highlighting strengths and weaknesses.

3. Multi-Language Support:

- Enable assessments and interfaces in multiple languages to cater to a diverse user base.

4. Proctoring Capabilities:

- Integrate AI-powered proctoring tools to monitor students during exams using webcams and detect suspicious activities.

5. Gamification:

- Add gamified elements such as badges, leaderboards, and achievements to enhance user engagement and motivation.

6. Integration with External Systems:

- Link with Learning Management Systems (LMS) to provide seamless access to course materials and tests.
- Enable compatibility with third-party tools such as video conferencing platforms for live assessments.

7. Offline Functionality:

- Develop features allowing users to attempt assessments offline and sync their results when back online.

SKILL MATRIX

8. Mobile App Development:

- Create a dedicated mobile app to offer an optimized and convenient experience for smartphone users.

9. Dynamic Question Bank Management:

- Implement AI to generate new questions automatically based on syllabus updates and difficulty levels.

10. Blockchain for Certification:

- Use blockchain technology for secure and tamper-proof certificates and records.

These future requirements will ensure the system evolves to meet emerging needs, incorporates cutting-edge technologies, and maintains its relevance in a fast-changing digital landscape.

Objectives

The primary goal of this project is to address the limitations of the manual approach by systematically automating tasks currently performed manually. The project aims to focus on the following key aspects:

1. Automation of Non-Computerized Environments:

- Transform manual processes into streamlined, automated workflows.

2. Manpower Reduction:

- Minimize the dependency on human resources, reducing workload and operational costs.

3. Enhanced Processing Speed:

- Accelerate data processing and transactions to improve efficiency and save time.

4. Error Reporting and Handling:

- Include robust error detection and reporting mechanisms to notify users of any issues and provide guidance to resolve them. This will significantly reduce the error rate.

SKILL MATRIX

5. Comprehensive Reporting:

- Offer multiple options to generate various types of reports, catering to different analytical and operational needs.

6. Improved Security Features:

- Ensure data and system security by implementing features like password protection and secure access controls.

By achieving these objectives, the project ensures a modern, efficient, and user-friendly system that eliminates manual inefficiencies and delivers reliable performance.

Problem Definition: Skill Matrix System

The traditional Skill Matrix involves extensive paperwork and manual effort, often requiring significant human resources to administer and evaluate. This process can be time-consuming, error-prone, and inefficient. The objective of the **Skill Matrix System** is to eliminate these inefficiencies by automating the process and offering a faster, more scalable solution.

Key Problems with Conventional Skill Matrix:

1. **Manual Work:** The process involves filling out paper forms and manually evaluating the responses, which is time-consuming and prone to human error.
2. **Time Constraints:** The traditional method requires specific time slots for conducting Skill Matrix assessments, limiting flexibility.
3. **Geographical Restrictions:** Students or users must be present physically for the test, requiring significant logistical arrangements.
4. **Delayed Results:** Evaluation of answers is done manually, leading to a delay in the results.

Proposed Solution: Skill Matrix System

The Digital Skill Matrix System aims to automate the entire process of skill assessments. The solution involves the following key features:

1. **Elimination of Paperwork:** The system will store all skill assessment data digitally, reducing the need for paper and minimizing errors.

SKILL MATRIX

2. **Automated Evaluation:** Responses to questions (whether multiple-choice or descriptive) will be evaluated automatically, allowing for instant results. This will drastically reduce the time between submission and evaluation.
3. **24/7 Availability:** Unlike traditional systems that are time-bound, the Digital Skill Matrix can be conducted at any time, be it during the day or late at night. This increases the accessibility and flexibility for users.
4. **Remote Access:** Users can take the Skill Matrix from any location, ensuring that geographical restrictions do not hinder participation.
5. **Instant Results:** As the assessment is automated, results will be generated instantly, providing users with immediate feedback on their performance.

Benefits:

- **Efficiency:** Reduces time spent on manual evaluation and paperwork.
- **Scalability:** The system can handle a large number of users simultaneously, without the need for additional resources.
- **Flexibility:** Users can participate at any time and from any location, making the system more inclusive.
- **Accuracy:** Automation ensures that answers are evaluated accurately, eliminating the possibility of human error.
- **Speed:** Immediate generation of results allows for quick decision-making and feedback.

The **Skill Matrix System** is an innovative approach that simplifies skill assessments while providing users with a better experience. It enhances both the convenience of users and the efficiency of organizations conducting the assessments.

Purpose

The purpose of the **Skill Matrix System** is to streamline the process of conducting skill assessments for students by utilizing web-based technology. The system will allow students to complete their skill matrix assessments from the comfort of their own PCs, removing the need for physical paperwork and manual effort.

SKILL MATRIX

Key objectives of the system include:

- 1. Ease of Access:** Students can take the skill matrix assessments at their convenience, from anywhere, using a simple web interface on their personal computers.
- 2. Single-click Submission:** The system will allow students to answer questions and submit them with just one click, removing the complexity and time-consuming nature of manual submissions.
- 3. Paperless Process:** By digitizing the entire assessment process, the system eliminates the use of physical paper, reducing paper waste, the administrative burden, and the potential for human error.
- 4. Instant Feedback:** With automated evaluation, students will receive immediate results, helping them quickly gauge their skills and performance.

This system aims to improve efficiency, save time, and reduce the operational costs associated with traditional skill matrix assessments, while also offering a seamless and modern user experience.

Scope

The scope of the project is very vast. It can be further enhanced in which all India level Skill Matrix would be conducted. If the project is further enhanced and newer technologies, concepts and security measures would be implemented.

SKILL MATRIX

Chapter 2: Problem Formulation

Introduction

The **Problem Introduction** marks the beginning of the software development process. It serves as the foundation for identifying, understanding, and defining the core objectives that the system must fulfil. This stage is essential for ensuring that the software aligns with the client's needs and expectations.

Key Objectives of the Problem Introduction:

- 1. Defining the System's Purpose:** Clearly outline what the system is intended to do and how it will address the identified problems or requirements.
- 2. Client-Centric Approach:** The development process is driven by the client's needs, which often exist as broad ideas or expectations at the start. These needs must be carefully captured and translated into actionable requirements.
- 3. Requirement Gathering:** Effective communication with stakeholders in the client's organization is crucial to accurately understand and document their expectations and challenges.
- 4. Avoiding Ambiguity:** Ambiguities or misinterpretations in requirements can lead to dissatisfaction for the client and challenges for the developer. It is essential to achieve a thorough understanding to minimize communication gaps.

Importance of Accurate Requirement Analysis:

- Accurate and well-documented requirements are critical for the success of the software project. They ensure that:
 - The system analysis, design, and coding processes are based on a solid foundation.
 - The end product meets the expectations of the users.
 - Common pitfalls like misinterpretation, ambiguity, and mismatched expectations are avoided.
- **Consequences of Poor Analysis:** If the requirements are poorly understood or specified:

SKILL MATRIX

- The software may fail to meet user needs.
- The developer's efforts may be wasted, leading to frustration and additional costs.
- A well-designed interface or system appearance cannot compensate for fundamental functional deficiencies.

Process of Requirement Understanding:

1. **Initial Statement:** Start with a clear and concise task description, summarizing the system's main objective in a single sentence.
2. **Technical Specifications:** Follow up with a detailed, precise, and technically accurate explanation of the requirements.
3. **Collaborative Approach:** Regular communication and iteration between the client and developer to refine and validate the requirements.

The **Problem Introduction** phase is vital in setting the stage for all subsequent software development activities, including system analysis, design, and coding. By ensuring clarity and accuracy during this phase, the project is more likely to succeed in meeting its objectives and satisfying its stakeholders.

Feasibility Study

Every project, in theory, is feasible with infinite resources and time. However, real-world constraints such as limited resources, tight schedules, and budget constraints make it essential to evaluate the feasibility of a project early in the development process. A comprehensive feasibility study helps prevent wasted effort, financial loss, and professional embarrassment by ensuring a project is viable before committing significant resources to it.

The purpose of the feasibility study is to determine whether the proposed system is achievable and whether it should be pursued. Once feasibility is established, the analyst can proceed to prepare the detailed project specification, finalizing the project requirements. Feasibility studies are usually conducted under tight time constraints to ensure efficient decision-making.

The study can be categorized into the following dimensions:

SKILL MATRIX

Technical Feasibility

Technical feasibility evaluates whether the technical requirements of the project can be met with the available resources, including hardware, software, and technical expertise. It focuses on ensuring that the system can be implemented to satisfy user requirements effectively. Key technical considerations include:

- **Output Capabilities:** The system must efficiently produce required outputs, such as reports, data summaries, and communication.
- **Response Time:** Ensuring the system's response time under various conditions is minimal for smooth operation.
- **Transaction Processing:** The system must process a given volume of transactions at the required speed.
- **Data Communication:** The ability to share and transmit data between distinct locations seamlessly.

For the **Skill Matrix System**, the technical feasibility emphasizes system configuration rather than specific hardware models. This includes defining the number of required workstations and ensuring smooth interconnectivity for seamless operations and communication.

Operational Feasibility

Operational feasibility assesses whether the proposed system will function effectively in the real-world environment and meet organizational needs. This includes understanding whether the system aligns with the users' requirements and expectations. Critical questions for operational feasibility include:

- **Support and Acceptance:** Does the system have sufficient support from management and users? Resistance may arise if the current system is well-liked.
- **User Satisfaction:** Will the proposed system resolve issues with existing business methods or enhance operational efficiency?
- **Involvement:** Have users been involved in the planning and development process? Early involvement reduces resistance and ensures alignment with user needs.
- **Impact on Operations:** Will the system cause any disruptions or produce poor results in specific areas? Potential issues identified early can be addressed proactively.

SKILL MATRIX

- **Staff Performance:** Will staff performance be affected post-implementation? Minor issues at the initial stages could evolve into major problems later.

Economic Feasibility

Economic feasibility, also known as cost-benefit analysis, evaluates the financial viability of the project by comparing expected benefits and savings against associated costs. Key considerations include:

- **Cost vs. Benefits:** The system is considered economically feasible if its benefits outweigh its costs.
- **Savings and Investments:** Since the Skill Matrix System leverages existing infrastructure, the primary cost is the development of the software. No additional investments are required for implementation.
- **Ongoing Evaluation:** As the system progresses through its lifecycle, economic feasibility is revisited to ensure continued financial justification.

By minimizing manual effort and improving operational efficiency, the Skill Matrix System offers substantial cost savings over traditional methods.

Legal Feasibility

Legal feasibility ensures that the proposed system complies with relevant laws and regulations. It is essential to verify that:

- The system adheres to legal standards and guidelines.
- The concepts and ideas used in the system's design and implementation are not in violation of intellectual property rights.
- The system does not conflict with any legal obligations or organizational policies.

For the Skill Matrix System, legal feasibility confirms that the software adheres to all regulatory requirements and that its implementation and operation are legally sound.

Problem Specification

The problem lies in transitioning from a manual system to a fully automated, efficient system. The current manual approach has limitations, while the proposed technical system aims to address these issues.

SKILL MATRIX

Manual System

The manual system suffers from several drawbacks:

- **Time-Consuming:** The processes are slow and inefficient, often leading to delays.
- **Error-Prone:** Manual systems are susceptible to mistakes due to human intervention.
- **Unstructured Approach:** There is often no clear methodology to tackle problems, leading to inconsistent results.
- **Resource-Intensive:** Requires significant human effort and physical resources like paper.

Technical System

The technical system leverages modern tools and technologies to provide significant advantages over manual methods. It includes the use of computers, printers, FAX, and the Internet to streamline operations. Key features include:

- **Speed and Efficiency:** Processes are faster and more streamlined.
- **Accuracy:** Reduces human errors significantly, ensuring reliable outcomes.
- **User-Friendly Interfaces:** Simplifies operations for users with intuitive designs.
- **Reliability:** Offers robust systems that can handle large volumes of data.
- **Up-to-Date:** Keeps the system in line with the latest technological trends, avoiding obsolescence.

Need for a Skill Matrix System

The need for a **Skill Matrix System** arises from the limitations of traditional methods and the desire for enhanced performance through modern technology. This system incorporates Internet and web server technologies, enabling:

- **Browser-Based Access:** Users can access the system through web browsers on their devices.
- **Centralized Data Storage:** All data is stored on a web server, ensuring consistency and easy accessibility.

SKILL MATRIX

- **Remote Accessibility:** The system can be accessed from anywhere, allowing users to participate without geographical constraints.
- **Improved Collaboration:** Facilitates seamless sharing and updating of data in real time.

The Proposed System

The proposed system addresses the limitations of manual processes by providing a systematic, automated approach. Its key objectives include:

- **Automation of Non-Computerized Environments:** Transition manual processes to a digital framework.
- **Manpower Reduction:** Minimize the need for human resources, saving time and effort.
- **Enhanced Speed:** Accelerate data processing and transactions, reducing operational delays.
- **Error Reporting and Handling:** Include features to detect and alert users of potential errors, thereby reducing the overall error rate.
- **Report Generation:** Provide multiple options to generate detailed and accurate reports.
- **Advanced Security:** Implement robust security measures like password protection to ensure data confidentiality and integrity.

Modules Used in the Project

The project consists of two main modules: **Admin** and **Student**, each designed with specific features and functionalities to cater to their respective roles.

1. Admin Module

The **Admin** module manages the system, ensuring smooth operations and maintaining the platform.

Admin Features:

1. **View User Information:** The admin can access and review personal details of all registered users for management purposes.

2. **Add/Delete Questions:**

SKILL MATRIX

- Admins can create new questions to include in the exams.
- They can also remove outdated or irrelevant questions.

3. Manage News:

- Admins have the ability to post updates or news related to the project.
- They can delete any outdated or incorrect news.

2. Student Module

The **Student** module is for users who participate in skill assessments.

Student Features:

1. Online Exam Participation:

- Users can take the exam online, eliminating the need for physical setups.
- Questions are presented one by one, ensuring an organized examination process.

2. Question Navigation:

- Users can skip questions they are unsure about and return to them later.
- Users can provide answers to the questions at their own pace.

3. Post-Exam Review:

- After completing the exam, users can view a detailed analysis of their performance.
- Correct and incorrect answers are displayed to help users learn and improve.

4. Share Experience:

- Users can share feedback or their overall experience after viewing their results, encouraging engagement and improvements.

SKILL MATRIX

Existing System

The current system in use relies on manual processes, which present several limitations and challenges:

1. Manual Handling of Data:

- All details and records must be manually documented and managed.
- This method is time-consuming, inefficient, and prone to errors.

2. User Unfriendliness:

- The existing system lacks automation, making it less intuitive and user-friendly.
- Users and administrators face difficulty in navigating and managing the processes.

3. Error-Prone Processes:

- Due to the heavy reliance on manual work, the system is more susceptible to mistakes in data entry, processing, and reporting.

4. Limited Accessibility:

- The system does not leverage modern technologies such as the internet or centralized databases.
- Accessing and managing the system often requires physical presence, reducing flexibility for users.

5. Lack of Reporting and Analysis:

- Generating insights, reports, and analysis from the existing system is cumbersome and may not yield accurate results due to inconsistencies in data handling.

6. Inadequate Security:

- The system lacks robust security measures to protect sensitive data, leading to risks of unauthorized access or data breaches

SKILL MATRIX

Chapter 3: System Architecture and Design

Defining a System

A **system** can be defined as a collection of interconnected components that work collaboratively to achieve a specific objective. Each system operates by following a structured process, consisting of three primary components:

1. Input:

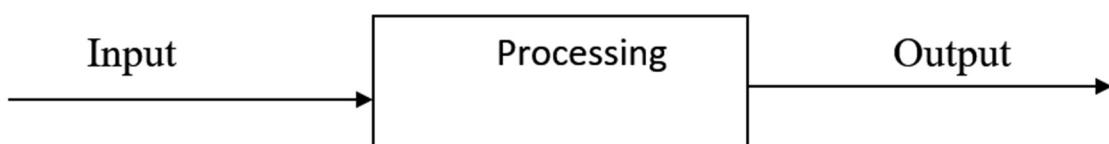
- **Definition:** This refers to the raw data or information provided to the system for processing.
- **Examples:**
 - In the context of the Skill Matrix system, inputs could include user registration details, exam questions, and user answers during the exam.

2. Processing:

- **Definition:** This is the phase where the input data is analyzed, transformed, or utilized to generate meaningful outcomes.
- **Examples:**
 - In the Skill Matrix system, processing involves validating user inputs, checking answers against correct solutions, and calculating scores.

3. Output:

- **Definition:** This is the final result or product of the system after processing the inputs.
- **Examples:**
 - Outputs in the Skill Matrix system include results of the skill exam, performance reports, and feedback for users.



SKILL MATRIX

In the **Skill Matrix system**, the components work as follows:

1. **Input:** User data, exam details, and user responses.
2. **Processing:** Exam evaluation, performance calculation, and report generation.
3. **Output:** Final scores, error notifications, and feedback.

System Development Life Cycle (SDLC)

The **System Development Life Cycle (SDLC)** is a comprehensive process that outlines the steps involved in developing, deploying, and maintaining an information system. It serves as a roadmap for developers and stakeholders, ensuring that the project meets its objectives efficiently and systematically. By following SDLC, developers can address problems proactively, minimize risks, and deliver high-quality solutions that meet user expectations.

Activities in the SDLC for the Project

1. Problem Definition

This initial phase focuses on identifying and defining the core problem the system aims to solve. It ensures a clear understanding of the objectives and establishes the foundation for subsequent phases.

- **Goal:** Answer the question, "*What problem are we solving, and why?*"

2. System Analysis

System analysis involves investigating the existing system, identifying inefficiencies, and gathering user requirements. This phase emphasizes understanding the "what" of the system rather than the "how."

- **Activities:**

- Conduct user interviews and surveys.
- Document existing workflows.
- Identify system bottlenecks and pain points.

3. Study of Existing System

This step examines the current system's operation, be it manual or semi-automated, to understand its structure, functionality, and limitations.

SKILL MATRIX

- **Outcome:** A comprehensive list of issues and potential improvements.

4. Drawbacks of the Existing System

An evaluation is performed to highlight inefficiencies, such as:

- Time consumption.
- Lack of scalability and flexibility.
- High error rates.
- Excessive reliance on manual intervention.

5. Proposed System

The proposed system is designed to address the identified drawbacks by leveraging technology to provide a faster, more reliable, and user-friendly solution.

- **Objectives:**

- Automate processes previously done manually.
- Enhance data accuracy and reduce error rates.
- Improve user satisfaction and accessibility.

6. System Requirement Study

This phase outlines the functional (what the system should do) and non-functional (performance, reliability, etc.) requirements.

- **Deliverables:**

- Use case diagrams.
- System requirement specifications (SRS).

7. Data Flow Analysis

A data flow analysis maps the movement of data within the system. Tools like **Data Flow Diagrams (DFDs)** illustrate how data is processed and stored.

- **Objective:** Ensure a smooth and logical data journey across all system modules.

SKILL MATRIX

8. Feasibility Study

A feasibility study evaluates whether the system is viable from technical, operational, economic, and legal perspectives.

1. Technical Feasibility:

- Assess the hardware, software, and network infrastructure required.
- Confirm compatibility with existing technologies.

2. Operational Feasibility:

- Evaluate user willingness and organizational readiness for the new system.
- Identify barriers to implementation and strategies to overcome them.

3. Economic Feasibility:

- Perform a cost-benefit analysis to determine if the system is financially justifiable.
- Estimate the return on investment (ROI).

4. Legal Feasibility:

- Ensure compliance with regulatory and legal standards.
- Verify that intellectual property and privacy concerns are addressed.

9. System Design

This phase transitions from "what" the system will do to "how" it will do it.

• Focus Areas:

- **Architectural Design:** Define the system's overall structure (e.g., client-server model).
- **Database Design:** Develop schemas, relationships, and data normalization.
- **Module Design:** Break down the system into manageable components.

SKILL MATRIX

10. Input Design (Database & Forms)

Input forms and database schemas are created to ensure accurate and efficient data entry and storage.

- **Tools:** ER diagrams, relational database management systems (RDBMS).

11. Updating

Plan mechanisms for regular updates, ensuring that the system remains relevant and functional over time.

12. Query/Report Design

Develop functionalities that allow users to generate customized reports and query data efficiently.

13. Administration

Set up administrative roles and permissions to manage user access, security, and data integrity.

14. Testing

Thorough testing ensures the system functions as intended under all conditions.

- **Types of Testing:** Unit testing, integration testing, system testing, and user acceptance testing (UAT).

15. Implementation

The system is deployed in the production environment, and users are trained to use it effectively.

16. Maintenance

Ongoing maintenance includes bug fixes, performance improvements, and adaptation to changing requirements.

System Analysis

System analysis involves studying existing systems, identifying problems, and recommending improvements. This phase ensures that the system aligns with user needs and organizational goals.

SKILL MATRIX

Objectives of System Analysis:

- Define clear requirements.
- Investigate system inefficiencies.
- Propose innovative solutions to meet user needs.

Steps in System Analysis:

1. **Requirements Gathering:** Use techniques like interviews, questionnaires, and observations.
2. **Problem Identification:** Analyze workflows and identify pain points.
3. **Proposed Solutions:** Recommend enhancements to address identified issues.

System Design

System design involves creating a detailed plan for implementing the system. It bridges the gap between requirements and development.

Key Elements of System Design:

- **Architecture:** High-level design defining system structure.
- **Components:** Detailed design of individual modules.
- **Interfaces:** Specifications for communication between components.
- **Data Design:** Schema definition for the database.

Modern Approaches:

- **Object-Oriented Design (OOD):** Focus on objects representing system entities.
- **Unified Modelling Language (UML):** Standardized visual modeling for complex systems.

By adhering to these SDLC activities and principles, the project is set to deliver a reliable, user-friendly, and efficient system while minimizing risks and ensuring stakeholder satisfaction.

SKILL MATRIX

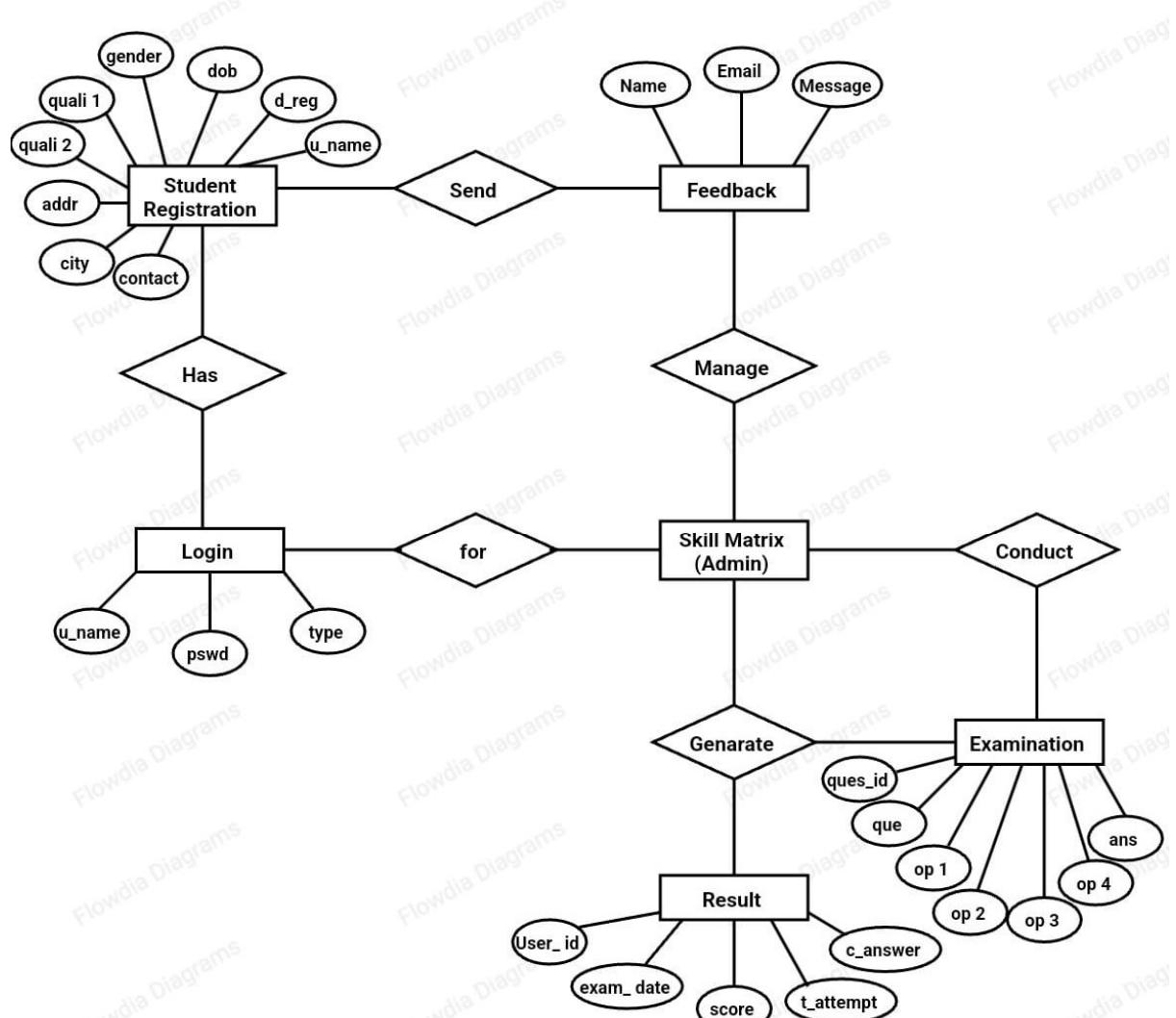
E-R Diagram

The entity-relationship model or entity-relationship diagram (ERD) is a data model or diagram for high-level descriptions of conceptual data model, and it provides a graphical notation for representing such data models in the form of entity-relationship diagrams. Such models are typically used in the first stage of information-system design; they are used, for Skill Matrix, to describe information needs and/or the type of information that is to be stored in the database during the requirement analysis. The data modelling technique, however, can be used to describe any ontology (i.e. an overview and classifications of used terms and their relationships) for a certain universe of discourse (i.e. area of interest).

In the case of the design of an information system that is based on a database, the conceptual data model is, at a later stage (usually called logical design), mapped to a logical data model, such as the relational model; this in turn is mapped to a physical model during physical design. Note that sometimes, both of these phases are referred to as "physical design". There are a number of conventions for entity-relationship diagrams (ERDs). The classical notation is described in the remainder of this article, and mainly relates to conceptual modelling. There are a range of notations more typically employed in logical and physical database design.

SKILL MATRIX

E-R Diagram



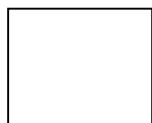
SKILL MATRIX

Expanded Explanation of Data Flow Diagram (DFD)

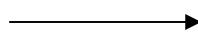
A **Data Flow Diagram (DFD)** is a graphical representation of the flow of data within a system. It provides a detailed view of how data is input, processed, stored, and output in a system, making it a critical tool in the design phase of software engineering. The primary goal of the DFD is to map out the system processes, data stores, and external entities to help in understanding, analyzing, and improving the system's functionality.

The DFD was first introduced by **Larry Constantine**, a prominent computer scientist, as a means of visualizing major transformations and flow of data in a system. It is also known as a **Bubble Chart** because of its use of circles (or "bubbles") to represent processes.

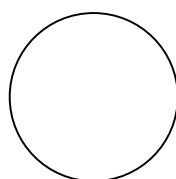
1. A square defines a source (originator) or destination of system data.



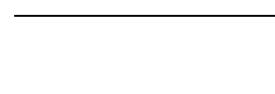
2. An arrow identifies data flow-data in motion. It is a pipeline through which information flows.



3. A circle or a "bubble" (Some people use an oval bubble) represents a process that transfers incoming data flows into outgoing data flows.



4. An open rectangle is a data store-data at rest, or a temporary repository of data.

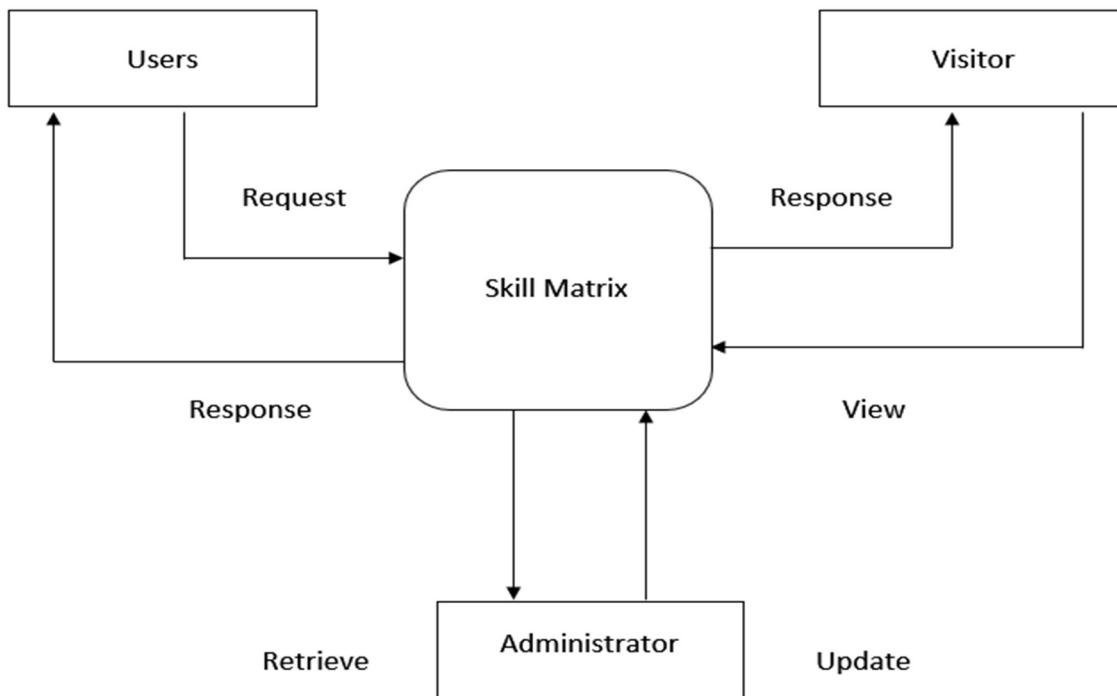


SKILL MATRIX

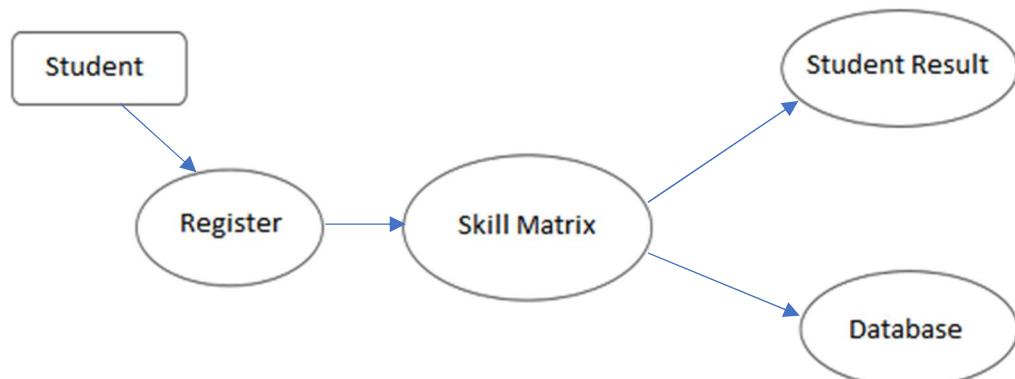
Levels of DFD

The DFD is typically created in multiple levels to provide varying degrees of detail:

1. **Context Diagram (Level 0 DFD):** This is the highest-level DFD, providing an overview of the entire system. It represents the system as a single process and shows its interactions with external entities.



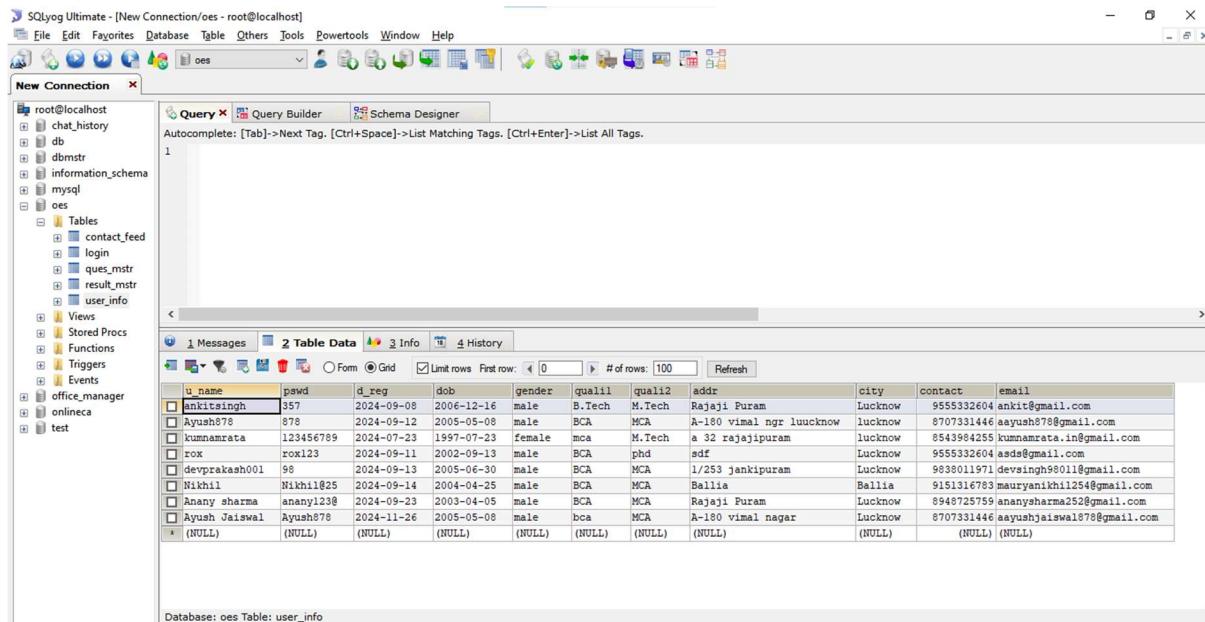
2. **Level 1 DFD:** This level breaks down the single process in the context diagram into sub-processes, showing more detail about the system's internal data flows.



SKILL MATRIX

Data Tables

User Information Table:



The screenshot shows the SQLyog Ultimate interface with a connection to 'oes' database. The left sidebar shows the database structure with 'Tables' expanded, displaying 'user_info'. The main area shows the 'Table Data' tab for the 'user_info' table. The table has the following columns: u_name, pswd, d_reg, dob, gender, qual1, qual12, addr, city, contact, and email. The data grid displays 10 rows of user information.

u_name	pswd	d_reg	dob	gender	qual1	qual12	addr	city	contact	email
ankitsingh	357	2024-09-08	2006-12-16	male	B.Tech	M.Tech	Rajaji Puram	Lucknow	9555332604	ankit@gmail.com
Ayush878	878	2024-09-12	2005-05-08	male	BCA	MCA	A-180 vimal ngr luucknow	lucknow	8707331446	ayush878@gmail.com
kumnamrata	123456789	2024-07-23	1997-07-23	female	mca	M.Tech	a 32 rajajipuram	lucknow	8543984255	kumnamrata.in@gmail.com
rox	rox123	2024-09-11	2002-09-13	male	BCA	phd	sdf	Lucknow	9555332604	ads@gmail.com
devprakash001	99	2024-09-13	2005-06-30	male	BCA	MCA	1/253 jankipuram	Lucknow	9838011971	devsingh99011@gmail.com
Nikhil	Nikhil825	2024-09-14	2004-04-25	male	BCA	MCA	Ballia	Ballia	9151316793	mauryanikhil254@gmail.com
Anany sharma	anany123@	2024-09-23	2003-04-05	male	BCA	MCA	Rajaji Puram	Lucknow	8548725759	ananysharma252@gmail.com
Ayush Jaiswal	Ayush878	2024-11-26	2005-05-08	male	bca	MCA	A-180 vimal nagar	Lucknow	8707331446	aayushjaiswal878@gmail.com
(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)

SKILL MATRIX

Login Table:

The screenshot shows the SQLyog Ultimate interface. The left sidebar displays the database structure under 'oes'. The main area has tabs for 'Query', 'Query Builder', and 'Schema Designer'. The 'Query' tab contains the following SQL code:

```
1
```

The 'Table Data' tab shows the contents of the 'login' table:

u_name	pswd	type
admin	admin	Admin
ankitsingh	357	Student
Ayush Jaishwal	Ayush878	Student
Ayush878	878	Student
devprakash001	dev	Student
kumnamrata	123456789	Student
Nikhil	Nikhil@25	Student
rox	rox123	Student
*	(NULL)	(NULL)

Below the table, the status bar indicates 'Database: oes Table: login'.

SKILL MATRIX

Questions Table:

The screenshot shows the SQLyog Ultimate interface with a database connection to 'oes' on 'localhost'. The left sidebar displays the database schema with tables like 'contact_feed', 'login', 'ques_mstr', 'result_mstr', and 'user_info'. The main window shows a table named 'ques_mstr' with 25 rows of data. The columns are 'ques_id', 'que', 'op1', 'op2', 'op3', and 'op4'. The data consists of multiple-choice questions and their corresponding options.

ques_id	que	op1	op2	op3	op4
1	Which of the following is a valid declaration of a float in Java?	float f = 5.0;	float f = 5;	float f = 5.0f;	Both B and C
2	Which of the following keywords is used to create a class in Java?	new	class	create	object
3	Which of the following is not a Java primitive data type?	int	float	String	boolean
4	What does JVM stand for?	Java Virtual Machine	Java Variable Machine	Java Visual Machine	Java Version
5	Which of the following methods can be used to read input from the console in Java?	System.in	Scanner	InputStream	Both A and B
6	Which operator is used to compare two values in Java?	=	==	!=	for-loop
7	Which of the following is used to handle exceptions in Java?	try-catch	if-else	switch-case	.PY
8	What is the default value of a boolean variable in Java?	true	false	0	null
9	What is the correct way to declare an array in Java?	int arr[] = new int[5];	int arr = new int[5];	int[] arr = int[5];	-P
10	What is the correct file extension for Python files?	.pyth	.python	.PY	.P
11	Which of the following is a valid way to declare a variable in Python?	int x = 10	x = 10	var x=10	x: int = 10
12	Which keyword is used to define a function in Python?	def	func	define	function
13					
14					
15					
16					
17					
18					
19					
20					
21					
22					
23					
24					
25					

SKILL MATRIX

Result Table:

The screenshot shows the SQLyog Ultimate interface with a connection to the 'oes' database. The left sidebar displays the database schema, including the 'oes' database which contains tables like 'contact_feed', 'login', 'ques_mstr', 'result_mstr', and 'user_info'. The main area shows the 'Query' tab with the following SQL query:

```
1
```

The results pane displays the data from the 'result_mstr' table:

user_id	exam_date	score	t_attempt	c_answer
AYUSH JAISWAL	2024-11-26	0	0	0
devprakash001	2024-11-26	0	0	0
Nikhil	2024-09-14	10	26	10
(NULL)	(NULL)	(NULL)	(NULL)	(NULL)

Below the results, the status bar indicates "Database: oes Table: result_mstr".

SKILL MATRIX

Contact US Table:

The screenshot shows the SQLyog Ultimate interface. On the left, the database structure is displayed under the 'oes' database. Under 'Tables', the 'contact_feed' table is selected. In the main pane, the 'Table Data' tab is active, showing the following data:

name	email	message
Dev Prakash Singh	devsingh98011@gmail.com	This is a very good website
Ragini	ragini@gmail.com	Excellent website
Nandini Pandey	nandini@gmail.com	Cool
(NULL)	(NULL)	(NULL)

Below the table, the status bar indicates "Database: oes Table: contact_feed".

SKILL MATRIX

Chapter 4: Testing and Implementation

Objective of Testing

Software testing is an investigation conducted to provide stakeholders with information about the quality of the product or service under test. Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Test techniques include, but are not limited to, the process of executing a program or application with the intent of finding software bugs (errors or other defects).

Software testing can be stated as the process of validating and verifying that a software program/application/product:

1. meets the requirements that guided its design and development;
2. works as expected; and
3. can be implemented with the same characteristics.

Software testing, depending on the testing method employed, can be implemented at any time in the development process. However, most of the test effort traditionally occurs after the requirements have been defined and the coding process has been completed having been shown that fixing a bug is less expensive when found earlier in the development process. Although in the Agile approaches most of the test effort is, conversely, on-going. As such, the methodology of the test is governed by the software development methodology adopted.

Different software development models will focus the test effort at different points in the development process. Newer development models, such as Agile, often employ test driven development and place an increased portion of the testing in the hands of the developer, before it reaches a formal team of testers. In a more traditional model, most of the test execution occurs after the requirements have been defined and the coding process has been completed. Testing can never completely identify all the defects within software. Instead, it furnishes a *criticism* or *comparison* that compares the state and behaviour of the product against oracles—principles or mechanisms by which someone might recognize a problem. These oracles may include (but are not limited to) specifications,

SKILL MATRIX

contracts, comparable products, past versions of the same product, inferences about intended or expected purpose, user or customer expectations, relevant standards, applicable laws, or other criteria.

Every software product has a target audience. For Skill Matrix, the audience for video game software is completely different from banking software. Therefore, when an organization develops or otherwise invests in a software product, it can assess whether the software product will be acceptable to its end users, its target audience, its purchasers, and other stakeholders. **Software testing** is the process of attempting to make this assessment.

Types of Testing

Black Box Testing

Black-box testing treats the software as a "black box"—without any knowledge of internal implementation. Black-box testing methods include: equivalence partitioning, boundary value analysis, all-pairs testing, fuzz testing, model-based testing, exploratory testing and specification-based testing.

- **Specification-based testing:** Specification-based testing aims to test the functionality of software according to the applicable requirements. Thus, the tester inputs data into, and only sees the output from, the test object. This level of testing usually requires thorough test cases to be provided to the tester, who then can simply verify that for a given input, the output value (or behaviour), either "is" or "is not" the same as the expected value specified in the test case.
Specification-based testing is necessary, but it is insufficient to guard against certain risks.
- **Advantages and disadvantages:** The black-box tester has no "bonds" with the code, and a tester's perception is very simple: a code *must* have bugs. Using the principle, "Ask and you shall receive," black-box testers find bugs where programmers do not. On the other hand, black-box testing has been said to be "like a walk in a dark labyrinth without a flashlight," because the tester doesn't know how the software being tested was actually constructed. As a result, there are situations when (1) a tester writes many

SKILL MATRIX

test cases to check something that could have been tested by only one test case, and/or
(2) some parts of the back-end are not tested at all.

Therefore, black-box testing has the advantage of "an unaffiliated opinion", on the one hand, and the disadvantage of "blind exploring", on the other

White Box Testing

White-box testing is when the tester has access to the internal data structures and algorithms including the code that implements these.

Types of white-box testing

The following types of white-box testing exist:

- API testing (application programming interface) - testing of the application using public and private APIs
- Code coverage - creating tests to satisfy some criteria of code coverage (e.g., the test designer can create tests to cause all statements in the program to be executed at least once)
- Fault injection methods - improving the coverage of a test by introducing faults to test code paths
- Mutation testing methods
- Static testing - All types

Test coverage

White-box testing methods can also be used to evaluate the completeness of a test suite that was created with black-box testing methods. This allows the software team to Skill Matrix parts of a system that are rarely tested and ensures that the most important function points have been tested.^[21]

Two common forms of code coverage are:

- *Function coverage*, which reports on functions executed
- *Statement coverage*, which reports on the number of lines executed to complete the test

They both return a code coverage metric, measured as a percentage.

SKILL MATRIX

Functional Testing

Functional testing refers to activities that verify a specific action or function of the code. These are usually found in the code requirements documentation, although some development methodologies work from use cases or user stories. Functional tests tend to answer the question of "can the user do this" or "does this particular feature work."

Non-functional testing refers to aspects of the software that may not be related to a specific function or user action, such as scalability or other performance, behavior under certain constraints, or security. Testing will determine the flake point, the point at which extremes of scalability or performance leads to unstable execution. Non-functional requirements tend to be those that reflect the quality of the product, particularly in the context of the suitability perspective of its users.

System Testing

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black box testing, and as such, should require no knowledge of the inner design of the code or logic.

As a rule, system testing takes, as its input, all of the "integrated" software components that have successfully passed integration testing and also the software system itself integrated with any applicable hardware system(s). The purpose of integration testing is to detect any inconsistencies between the software units that are integrated together (called *assemblages*) or between any of the *assemblages* and the hardware. System testing is a more limited type of testing; it seeks to detect defects both within the "inter-assemblages" and also within the system as a whole.

Various level Of Testing

Before implementation the system is tested at two levels:

1. Level 1
2. Level 2

SKILL MATRIX

Level 1 Testing (Alpha Testing)

Alpha testing is simulated or actual operational testing by potential users/customers or an independent test team at the developers' site. Alpha testing is often employed for off-the-shelf software as a form of internal acceptance testing, before the software goes to beta testing.

Level 2 Testing (Beta testing)

Beta testing comes after alpha testing and can be considered a form of external user acceptance testing. Versions of the software, known as beta versions, are released to a limited audience outside of the programming team. The software is released to groups of people so that further testing can ensure the product has few faults or bugs. Sometimes, beta versions are made available to the open public to increase the feedback field to a maximal number of future users.

Recovery and Security

A forced system failure is induced to test a backup recovery procedure for file integrity. Inaccurate data are entered to see how the system responds in terms of error detection and protection. Related to file integrity is a test to demonstrate that data and programs are secure from unauthorized access.

Usability Documentation and Procedure:

The usability test verifies the user-friendly nature of the system. This relates to normal operating and error-handling procedures.

Quality Assurance

Though controversial, software testing is a part of the software quality assurance (SQA) process. In SQA, software process specialists and auditors are concerned for the software development process rather than just the artifacts such as documentation, code and systems. They Skill Matrix and change the software engineering process itself to reduce the number of faults that end up in the delivered software: the so-called defect rate.

What constitutes an "acceptable defect rate" depends on the nature of the software; A flight simulator video game would have much higher defect tolerance than software for an actual airplane.

Although there are close links with SQA, testing departments often exist independently, and there may be no SQA function in some companies.

SKILL MATRIX

Software testing is a task intended to detect defects in software by contrasting a computer program's expected results with its actual results for a given set of inputs. By contrast, QA (quality assurance) is the implementation of policies and procedures intended to prevent defects from occurring in the first place.

System Implementation

During the implementation stage the system is physically created. Necessary programs are coded, debugged and documented. A new hardware is selected, ordered and installed.

System Specification

Every computer system consists of three major elements.

1. The Hardware
2. Application software such as visual studio
3. Operating system

For successful operation of the package following must be kept in mind:

Too many packages should not be used, as very few system may have all those packages installed due to memory problem. Thus, the compatibility of the system development will get reduced.

Hardware Requirements

- Pentium IV Processor
- 512 MB RAM
- 40 GB HDD
- Color Monitor
- Keyboard, Mouse

Software Requirements

Front-end Tool: - JSP, HTML, JAVASCRIPT.

- User friendly
- Low-Cost Solution
- GUI feature
- Better designing aspects.

SKILL MATRIX

Back-end Tool: - SQLyog

Features are:

- The graphical management tools make it extremely easy to manage the server.
- Security
- Portability
- Performance
- Scalability
- Reliability

Installation

The application installation script has to be generated from the current server where the application source code saved and installed in the main server from where the application is to be run. This was done using a special code, which generates all SQL-Statements to insert preliminary data (like menu entries, code in code directories etc) at server and the operational modules of the application made available to the end user successfully.

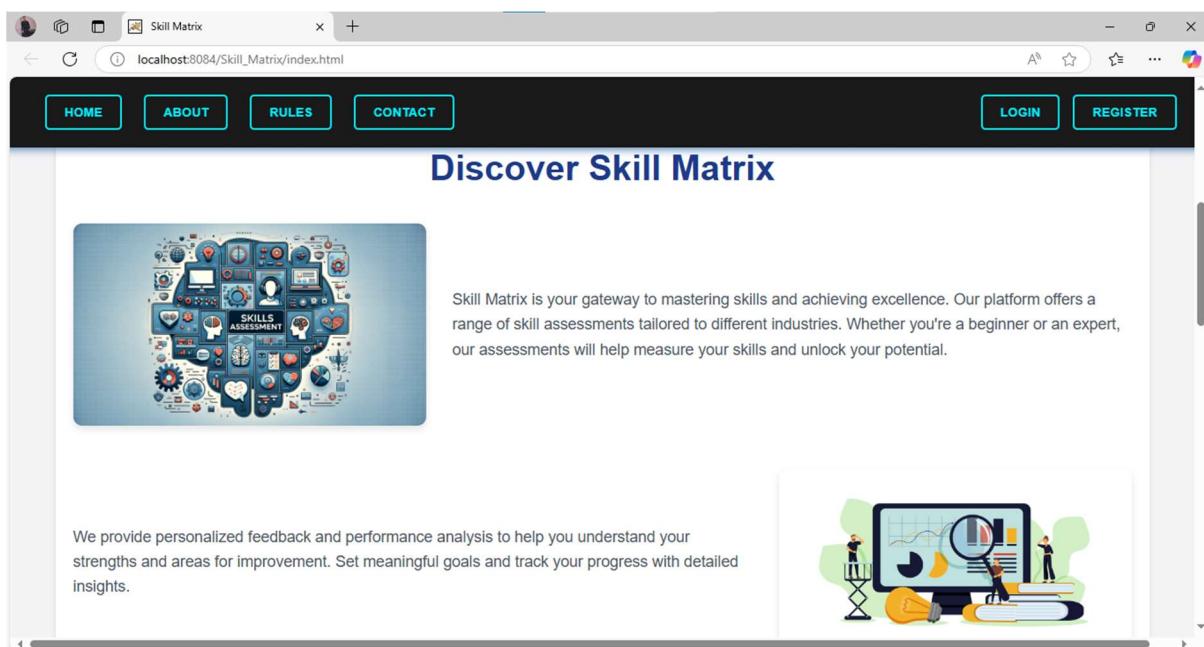
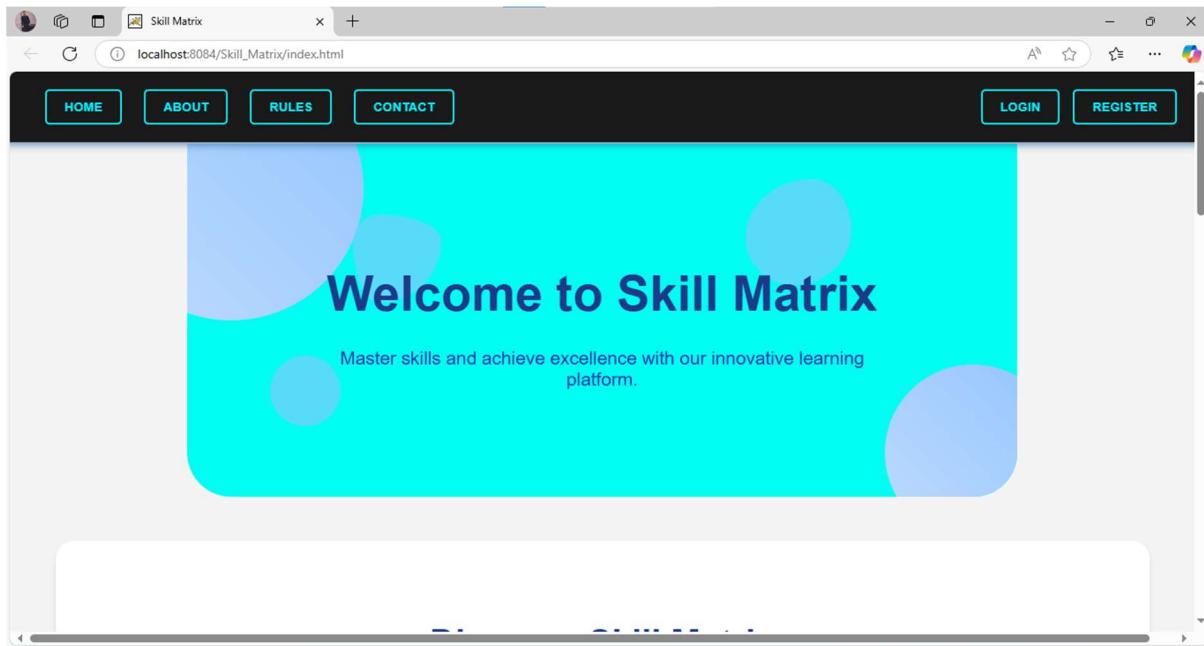
Implementation

The system is still under construction few reports are yet to me made after that this system will be implanted at client side. Users will be given a training to use the package and special work shop is conducted by the courier for the purpose and according to their feedback the change implanted in the software.

SKILL MATRIX

Screen Shots

Home Page:



SKILL MATRIX

The screenshot shows the homepage of the Skill Matrix platform. At the top, there is a navigation bar with links for HOME, ABOUT, RULES, CONTACT, LOGIN, and REGISTER. Below the navigation bar, there is a graphic of four stylized human figures with colored arrows pointing towards graduation caps, symbolizing learning paths. To the right of this graphic, a text block states: "Our platform offers flexible learning paths, allowing you to pursue self-paced learning or structured courses that suit your needs, ensuring continuous growth and motivation." Below this text, another section encourages users to join a community of learners and professionals, with a subtext: "Skill Matrix connects you with like-minded individuals passionate about self-improvement." To the right of this text is a graphic featuring a grid of colored squares and several overlapping circles representing various professional skills like Business Intelligence, Big Data, Analytics, Project Management, etc.

The screenshot shows the homepage of the Skill Matrix platform, specifically focusing on its features. The title "Our Features" is centered at the top. Below it are three main feature sections: 1) "Real-Time Assessments" with a subtext explaining immediate feedback and real-time data analysis. 2) "Personalized Learning Paths" with a circular diagram showing five methods: Resume reviews, Skill assessment tests, Job simulations, Reference checks, and Job interviews. 3) "Collaborative Environment" with a subtext encouraging engagement with a vibrant community of professionals and learners. Each feature section includes a small image icon.

SKILL MATRIX

About US Page:

Welcome to Skill Matrix, the ultimate platform to enhance your skills and unlock your full potential. We are committed to providing innovative tools and resources for learners and professionals alike.



Why Choose Skill Matrix?

Comprehensive Assessments
Evaluate your skills with our tailored assessments designed for every industry.

Personalized Learning
Enjoy flexible learning paths to match your unique goals and schedule.

Collaborative Community
Engage with like-minded individuals to share ideas and grow together.

© 2024 Skill Matrix. All rights reserved.

SKILL MATRIX

Rules Page:

The screenshot shows a web browser window titled "Rules - Skill Matrix" with the URL "localhost:8084/Skill_Matrix/rules.jsp". The page has a dark header bar with white text and buttons for "HOME", "ABOUT", "RULES", "CONTACT", "LOGIN", and "REGISTER". The main title "Rules of Skill Matrix" is centered above a paragraph of text. Below the paragraph are five bullet points. At the bottom of the page is a copyright notice.

To maintain a smooth and ethical experience on our platform, please follow the guidelines below:

- Integrity and Honesty:** Always act with integrity during assessments.
- No Unauthorized Materials:** Using unauthorized resources is strictly prohibited.
- Time Management:** Adhere to the time limits for each activity.
- Follow Instructions:** Carefully follow all provided guidelines.
- Report Issues:** Immediately report any technical problems or concerns.

© 2024 Skill Matrix. All rights reserved. | [Contact Us](#)

Contact US Page:

The screenshot shows a web browser window titled "Contact Us - Skill Matrix" with the URL "localhost:8084/Skill_Matrix/contact.jsp". The page has a dark header bar with white text and buttons for "HOME", "ABOUT", "RULES", "CONTACT", "LOGIN", and "REGISTER". The main title "Contact Us" is centered above a paragraph of text. Below the paragraph is a form with three input fields: "Your Name", "Your Email", and "Your Message". A large blue "Send Message" button is at the bottom of the form.

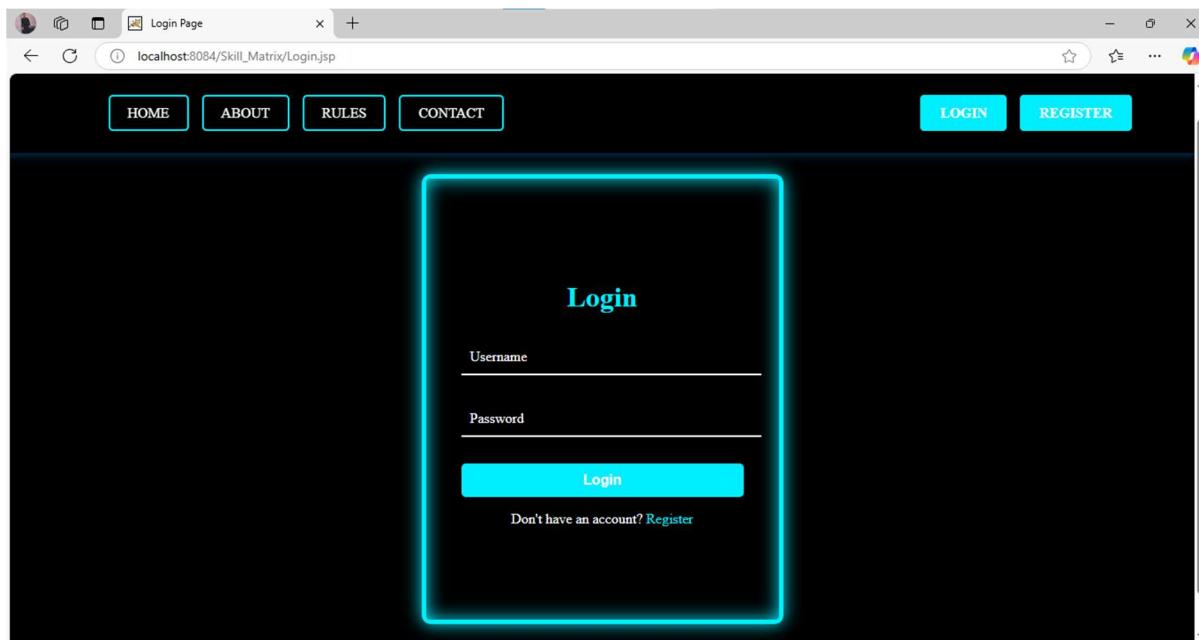
We'd love to hear from you! Whether you have questions, suggestions, or just want to say hello, please reach out to us using the form below.

Your Name
Your Email
Your Message

Send Message

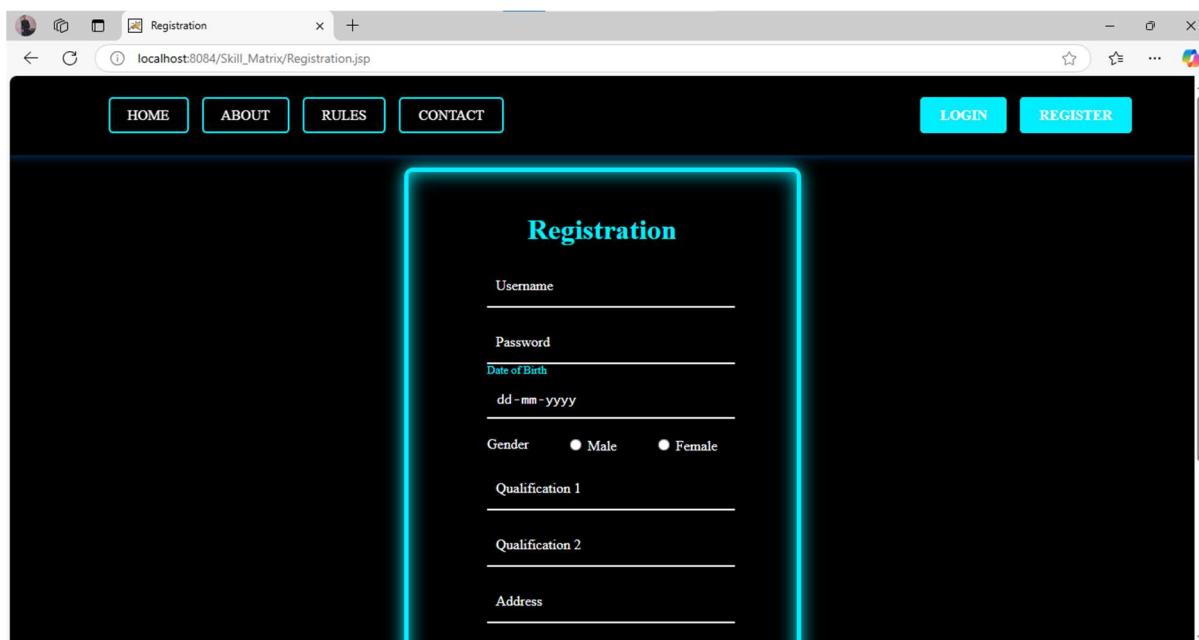
SKILL MATRIX

Login Page:



A screenshot of a web browser window titled "Login Page". The URL is "localhost:8084/Skill_Matrix/Login.jsp". The page has a dark background with a central light blue rectangular form. At the top of the form is the word "Login" in bold. Below it are two input fields: "Username" and "Password", each with a horizontal line underneath. Underneath the password field is a blue rectangular button with the word "Login" in white. Below the button is a small line of text that says "Don't have an account? [Register](#)". At the very bottom of the form is a horizontal scroll bar.

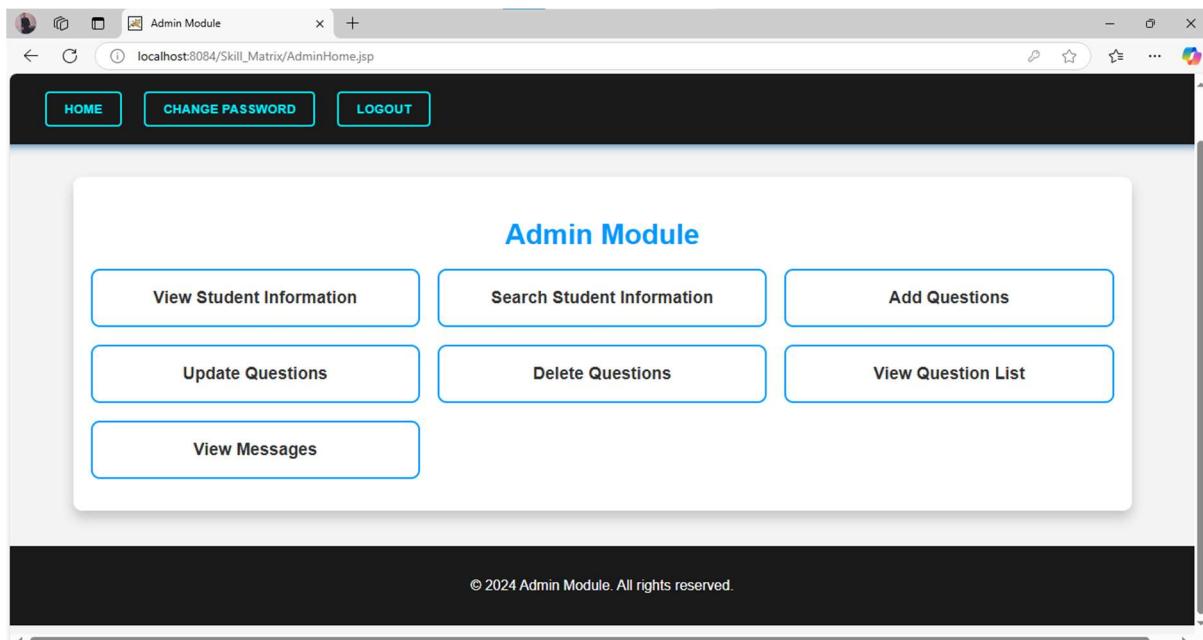
Registration Page:



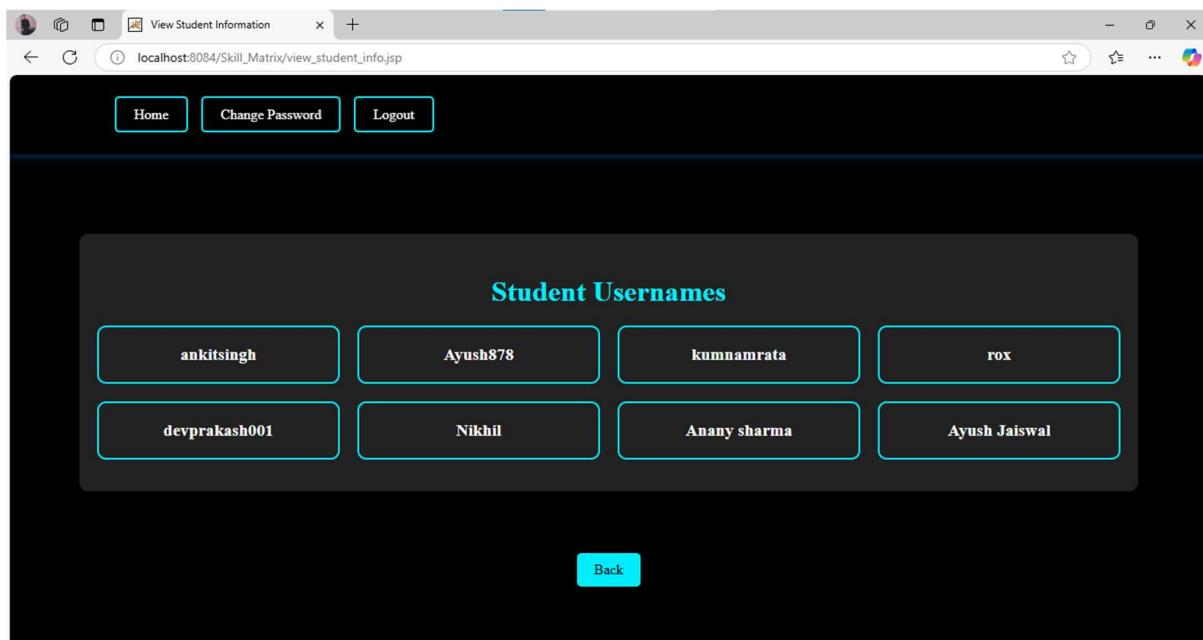
A screenshot of a web browser window titled "Registration". The URL is "localhost:8084/Skill_Matrix/Registration.jsp". The page has a dark background with a central light blue rectangular form. At the top of the form is the word "Registration" in bold. Below it are several input fields: "Username", "Password", and "Date of Birth" (with a placeholder "dd-mm-yyyy"). There is also a gender selection section with the words "Gender" and two radio buttons: one for "Male" and one for "Female", both of which are currently selected. Below these are two more input fields for "Qualification 1" and "Qualification 2", followed by an "Address" input field. At the bottom of the form is a horizontal scroll bar.

SKILL MATRIX

Admin Home Page:

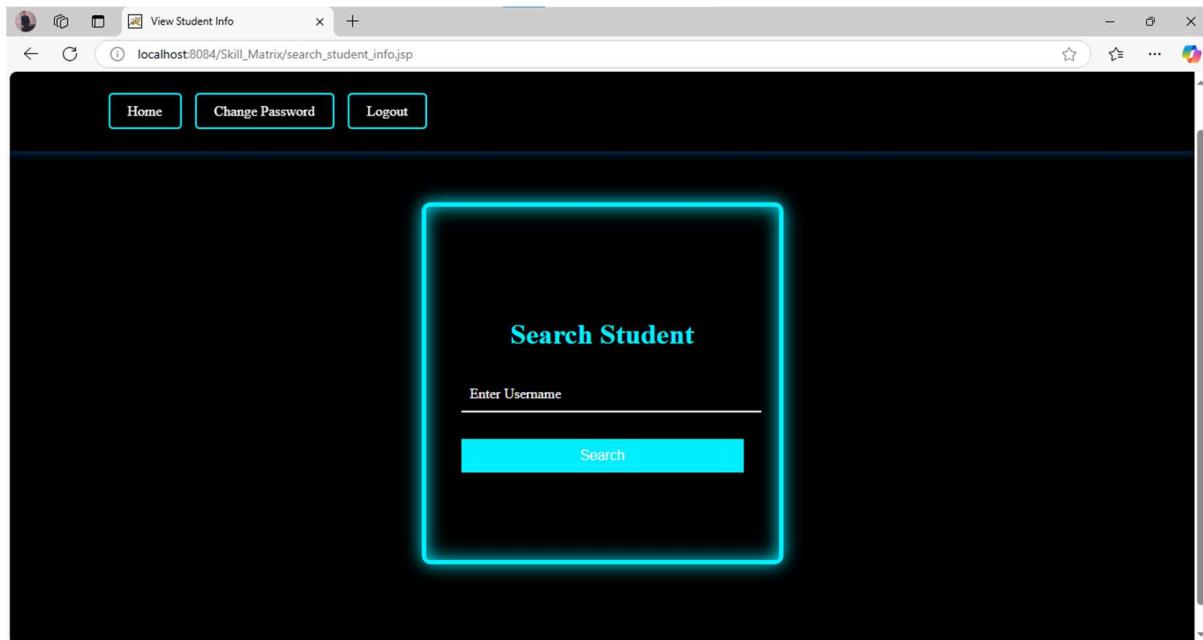


View Student Page:



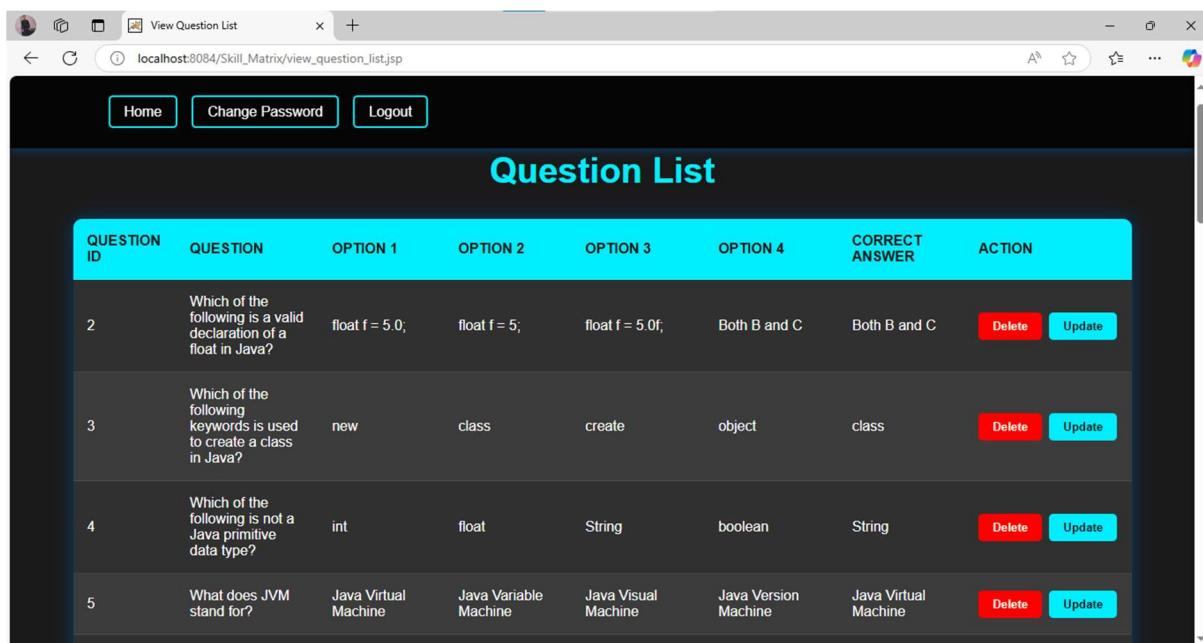
SKILL MATRIX

Search Student Page:



The screenshot shows a web browser window with the URL `localhost:8084/Skill_Matrix/search_student_info.jsp`. The page has a dark theme with a red header bar containing 'Home', 'Change Password', and 'Logout' buttons. Below the header is a large red-bordered input field labeled 'Search Student'. Inside this field, there is a smaller input field labeled 'Enter Username' and a blue 'Search' button.

Add Question Page:



The screenshot shows a web browser window with the URL `localhost:8084/Skill_Matrix/view_question_list.jsp`. The page has a dark theme with a red header bar containing 'Home', 'Change Password', and 'Logout' buttons. Below the header is a title 'Question List' and a table with the following data:

QUESTION ID	QUESTION	OPTION 1	OPTION 2	OPTION 3	OPTION 4	CORRECT ANSWER	ACTION
2	Which of the following is a valid declaration of a float in Java?	float f = 5.0;	float f = 5;	float f = 5.0f;	Both B and C	Both B and C	<button>Delete</button> <button>Update</button>
3	Which of the following keywords is used to create a class in Java?	new	class	create	object	class	<button>Delete</button> <button>Update</button>
4	Which of the following is not a Java primitive data type?	int	float	String	boolean	String	<button>Delete</button> <button>Update</button>
5	What does JVM stand for?	Java Virtual Machine	Java Variable Machine	Java Visual Machine	Java Version Machine	Java Virtual Machine	<button>Delete</button> <button>Update</button>

SKILL MATRIX

Messages from Contact Form Page:

The screenshot shows a web browser window titled "View Contact Messages - Admin" with the URL "localhost:8084/Skill_Matrix/view_messages.jsp". The page has a dark theme with a header containing "Home", "Change Password", and "Logout" buttons. Below the header is a section titled "Messages from Contact Form" which contains a table with three rows of data. The table has columns for "Name", "Email", and "Message".

Name	Email	Message
Dev Prakash Singh	devsingh98011@gmail.com	This is a very good website
Ragini	ragini@gmail.com	Excellent website
Nandini Pandey	nandini@gmail.com	Cool

A "Back" button is located at the bottom center of the page.

Student Home Page:

The screenshot shows a web browser window titled "Student Module" with the URL "localhost:8084/Skill_Matrix/StudentHome.jsp". The page has a dark theme with a header containing "HOME", "CHANGE PASSWORD", and "LOGOUT" buttons. Below the header is a central box titled "Student Module" containing five buttons arranged in two rows: "Profile", "Edit Profile", "Start Exam" in the top row, and "View Result", "Delete Your Account" in the bottom row.

At the bottom of the page, there is a black footer bar with the text "© 2024 Student Module. All rights reserved."

SKILL MATRIX

Your Profile Page:

The screenshot shows a web browser window titled "View Profile" with the URL "localhost:8084/Skill_Matrix/ViewProfile.jsp". The page has a dark header with buttons for "Home", "Edit Profile", "Change Password", and "Logout". Below the header is a section titled "Your Profile" containing a table with the following data:

Field	Value
Username	devprakash001
Date of Birth	2005-06-30
Gender	male
Qualification 1	BCA
Qualification 2	MCA
Address	1/253 jankipuram
City	Lucknow
Contact	9838011971

Exam Page:

The screenshot shows a web browser window titled "Start Exam" with the URL "localhost:8084/Skill_Matrix/Exam.jsp". The page has a dark header with buttons for "HOME", "CHANGE PASSWORD", and "LOGOUT". Below the header is a section titled "Start Exam" with a red "Time Left: 00:09:58". A question box contains the following text:

Question 1: Which of the following is a valid declaration of a float in Java?

float f = 5.0;
 float f = 5;
 float f = 5.0f;
 Both B and C

At the bottom of the question box are "Previous" and "Next" buttons.

SKILL MATRIX

Submit Page:

The screenshot shows a web browser window titled "Exam Result" with the URL "localhost:8084/Skill_Matrix/SubmitExam.jsp". The page displays a score of "Your Score: 23 out of 25". Below the score, there is a "Questions Summary" section containing five questions with their respective answers and correct answers.

Question	Your Answer	Correct Answer
Which of the following is a valid declaration of a float in Java?	Both B and C	Both B and C
Which of the following keywords is used to create a class in Java?	class	class
Which of the following is not a Java primitive data type?	int	String
What does JVM stand for?	Java Virtual Machine	Java Virtual Machine
Which of the following methods can be used to read input from the console in Java?	Both A and B	Both A and B

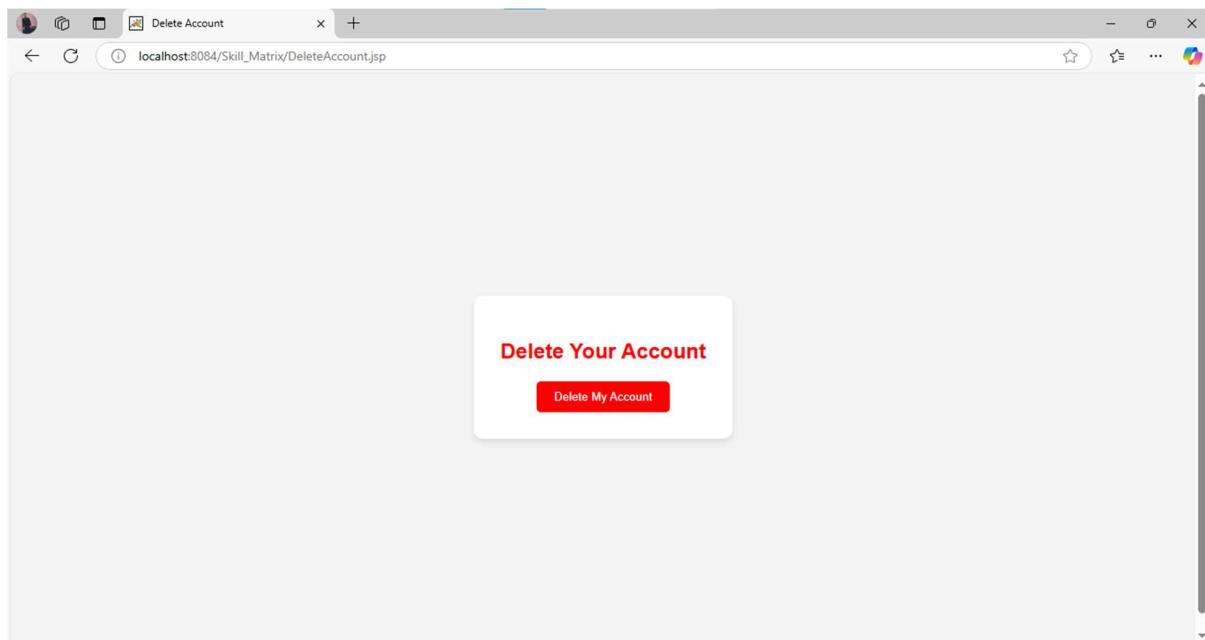
Result Page:

The screenshot shows a web browser window titled "View Exam Result" with the URL "localhost:8084/Skill_Matrix/ViewResult.jsp". The page features a navigation bar with "HOME", "CHANGE PASSWORD", and "LOGOUT" buttons. Below the navigation bar, there is a section titled "Your Exam Result" which contains a table summarizing the exam details.

Exam Date	Score	Total Attempts	Correct Answers
2024-11-28	23	25	23

SKILL MATRIX

Delete Account Page:



SKILL MATRIX

Coding

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Skill Matrix</title>

    <style>

        @import
url('https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;600;700&display=swap');

        body {

            font-family: 'Poppins', sans-serif;

            margin: 0;

            padding: 0;

            display: flex;

            flex-direction: column;

            align-items: center;

            background-color: #f4f4f4;

            color: #333;

        }

.neon-navbar {



    display: flex;
```

SKILL MATRIX

```
justify-content: space-between;  
align-items: center;  
padding: 15px 30px;  
background-color: #1a1a1a;  
box-shadow: 0 2px 10px rgba(0, 78, 146, 0.7);  
position: sticky;  
top: 0;  
width: 100%;  
z-index: 1000;  
}
```

```
.neon-nav-left,  
.neon-nav-right {  
    display: flex;  
    gap: 25px;  
    align-items: center;  
}
```

```
.neon-nav-right {  
    margin-right: 20px;  
    gap: 15px;  
}
```

```
.neon-nav-left li,
```

SKILL MATRIX

```
.neon-nav-right li {
```

```
    list-style: none;
```

```
}
```

```
.neon-nav-left a,
```

```
.neon-nav-right a {
```

```
    text-decoration: none;
```

```
    color: #0ef;
```

```
    padding: 10px 20px;
```

```
    border: 2px solid #0ef;
```

```
    border-radius: 5px;
```

```
    transition: all 0.3s ease;
```

```
    font-weight: 600;
```

```
    text-transform: uppercase;
```

```
    font-size: 0.9rem;
```

```
}
```

```
.neon-nav-left a:hover,
```

```
.neon-nav-right a:hover {
```

```
    background-color: #0ef;
```

```
    color: #1a1a1a;
```

```
    box-shadow: 0 0 10px #0ef, 0 0 20px #0ef, 0 0 30px #0ef;
```

```
}
```

SKILL MATRIX

```
.banner {  
    position: relative;  
    margin-top: 2px;  
    width: 70%;  
    height: 400px;  
    background-color: #00fff2;  
    overflow: hidden;  
    border-radius: 0 0 50px 50px;  
    margin-bottom: 50px;  
}  
  
.
```

```
.banner::before,  
.banner::after {  
    content: " ";  
    position: absolute;  
    background: linear-gradient(45deg, #BFDBFE, #93C5FD);  
    border-radius: 50%;  
}  
  
.
```

```
.banner::before {  
    width: 300px;  
    height: 300px;  
    top: -100px;  
    left: -100px;  
}
```

SKILL MATRIX

}

```
.banner::after {  
    width: 200px;  
    height: 200px;  
    bottom: -50px;  
    right: -50px;  
}
```

```
.content {  
    position: relative;  
    z-index: 1;  
    display: flex;  
    flex-direction: column;  
    align-items: center;  
    justify-content: center;  
    height: 100%;  
    text-align: center;  
    padding: 0 20px;  
}
```

```
.content h1 {  
    color: #1E3A8A;  
    font-size: 3.5rem;
```

SKILL MATRIX

```
margin-bottom: 0.5rem;  
font-weight: 700;  
}
```

```
.content p {  
  
color: #1E40AF;  
  
font-size: 1.3rem;  
  
max-width: 600px;  
  
}
```

```
.shape {  
  
position: absolute;  
  
background-color: #93C5FD;  
  
opacity: 0.6;  
  
}
```

```
.shape-1 {  
  
width: 100px;  
  
height: 100px;  
  
top: 20%;  
  
left: 20%;  
  
border-radius: 30% 70% 70% 30% / 30% 30% 70% 70%;  
  
}
```

SKILL MATRIX

```
.shape-2 {  
    width: 80px;  
    height: 80px;  
    bottom: 20%;  
    left: 10%;  
    border-radius: 50%;  
}  
  
.shape-3 {  
    width: 120px;  
    height: 120px;  
    top: 10%;  
    right: 20%;  
    border-radius: 63% 37% 54% 46% / 55% 48% 52% 45%;  
}  
  
/* Welcome Section */  
  
.welcome-section {  
    padding: 60px 20px;  
    background-color: #ffffff;  
    max-width: 1200px;  
    margin: 0 auto;  
    text-align: center;  
    border-radius: 20px;
```

SKILL MATRIX

```
  box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);  
}  
  
}
```

```
.welcome-section h2 {  
  font-size: 2.5rem;  
  color: #1E3A8A;  
  margin-bottom: 40px;  
  font-weight: 700;  
}
```

```
.welcome-content {  
  display: flex;  
  flex-direction: column;  
  gap: 50px;  
}
```

```
.welcome-block {  
  display: flex;  
  flex-direction: row;  
  align-items: center;  
  gap: 30px;  
  justify-content: space-between;  
}
```

SKILL MATRIX

```
.welcome-block:nth-child(left) {  
    flex-direction: row-reverse;  
}  
  
.welcome-img {  
    max-width: 400px;  
    border-radius: 10px;  
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);  
}  
  
.welcome-text {  
    flex: 1;  
    font-size: 1.1rem;  
    color: #4B5563;  
    line-height: 1.6;  
    text-align: left;  
}  
  
.features-section {  
    padding: 60px 20px;  
    background-color: #EFF6FF;  
    text-align: center;  
    margin-top: 50px;  
}
```

SKILL MATRIX

```
.features-section h2 {  
    font-size: 2.2rem;  
    margin-bottom: 40px;  
    color: #1E3A8A;  
    font-weight: 700;  
}  
  
}
```

```
.features {  
    display: flex;  
    flex-wrap: wrap;  
    gap: 40px;  
    justify-content: center;  
}  
  
}
```

```
.feature-item {  
    background-color: #fff;  
    padding: 30px;  
    border-radius: 15px;  
    box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);  
    max-width: 300px;  
    text-align: left;  
    transition: transform 0.3s ease;  
}  
  
}
```

SKILL MATRIX

```
.feature-item:hover {  
    transform: translateY(-10px);  
}  
  
{
```

```
.feature-item img {  
    width: 100%;  
    border-radius: 8px;  
    margin-bottom: 20px;  
}  
  
{
```

```
.feature-item h3 {  
    font-size: 1.5rem;  
    color: #1E3A8A;  
    margin-bottom: 15px;  
    font-weight: 600;  
}  
  
{
```

```
.feature-item p {  
    font-size: 1rem;  
    color: #4B5563;  
    line-height: 1.5;  
}  
  
{
```

SKILL MATRIX

```
footer {  
background-color: #1a1a1a;  
color: #fff;  
padding: 20px;  
text-align: center;  
margin-top: 40px;  
width: 100%;  
}  
  
@media (max-width: 768px) {  
    .welcome-block {  
        flex-direction: column !important;  
    }  
}
```

```
.welcome-img {  
    max-width: 100%;  
}
```

```
.content h1 {  
    font-size: 2.5rem;  
}  
  
.
```

```
.content p {  
    font-size: 1.1rem;  
}
```

SKILL MATRIX

```
}

}

</style>

</head>

<body>

<nav class="neon-navbar">

<ul class="neon-nav-left">

<li><a href="index.html">Home</a></li>

<li><a href="about.jsp">About</a></li>

<li><a href="rules.jsp">Rules</a></li>

<li><a href="contact.jsp">Contact</a></li>

</ul>

<ul class="neon-nav-right">

<li><a href="Login.jsp" class="neon-btn">Login</a></li>

<li><a href="Registration.jsp" class="neon-btn">Register</a></li>

</ul>

</nav>

<div class="banner">

<div class="shape shape-1"></div>

<div class="shape shape-2"></div>

<div class="shape shape-3"></div>

<div class="content">
```

SKILL MATRIX

<h1>Welcome to Skill Matrix</h1>

<p>Master skills and achieve excellence with our innovative learning platform.</p>

</div>

</div>

<section class="welcome-section">

<h2>Discover Skill Matrix</h2>

<div class="welcome-content">

<div class="welcome-block">

<div class="welcome-text">

<p>Skill Matrix is your gateway to mastering skills and achieving excellence. Our platform offers a range of skill assessments tailored to different industries. Whether you're a beginner or an expert, our assessments will help measure your skills and unlock your potential.</p>

</div>

</div>

<div class="welcome-block">

<div class="welcome-text">

<p>We provide personalized feedback and performance analysis to help you understand your strengths and areas for improvement. Set meaningful goals and track your progress with detailed insights.</p>

</div>

SKILL MATRIX

```
</div>

<div class="welcome-block">

    <div class="welcome-text">

        <p>Our platform offers flexible learning paths, allowing you to pursue self-paced
learning or structured courses that suit your needs, ensuring continuous growth and
motivation.</p>

    </div>

</div>

<div class="welcome-block">

    <div class="welcome-text">

        <p>Join our community of learners and professionals, collaborate, share resources,
and grow together. Skill Matrix connects you with like-minded individuals passionate about
self-improvement.</p>

    </div>

    </div>

</div>

</section>

<section class="features-section">

    <h2>Our Features</h2>

    <div class="features">

        <div class="feature-item">
```

SKILL MATRIX

```


<h3>Real-Time Assessments</h3>

<p>Receive immediate feedback after each skill test and track your performance with real-time data analysis.</p>

</div>

<div class="feature-item">



<h3>Personalized Learning Paths</h3>

<p>Custom-tailored learning journeys designed to fit your skill level and career goals.</p>

</div>

<div class="feature-item">



<h3>Collaborative Environment</h3>

<p>Engage with a vibrant community of professionals and learners to exchange knowledge and ideas.</p>

</div>

</div>

</section>
```

```
<footer>

<p>&copy; 2024 Skill Matrix. All rights reserved.</p>

</footer>
```

SKILL MATRIX

```
</body>

</html>

<%@ page contentType="text/html;charset=UTF-8" language="java" %>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Admin Module</title>

<style>

    @import

url('https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;600;700&display=swap');

    body {

        font-family: 'Poppins', sans-serif;

        margin: 0;

        padding: 0;

        display: flex;

        flex-direction: column;

        align-items: center;

        background-color: #f4f4f4;

        color: #333;

    }


```

SKILL MATRIX

```
.neon-navbar {  
    display: flex;  
    justify-content: space-between;  
    align-items: center;  
    padding: 15px 30px;  
    background-color: #1a1a1a;  
    box-shadow: 0 2px 10px rgba(0, 78, 146, 0.7);  
    position: sticky;  
    top: 0;  
    width: 100%;  
    z-index: 1000;  
}  
  
}
```

```
.neon-nav-left,  
.neon-nav-right {  
    display: flex;  
    gap: 25px;  
    align-items: center;  
}  
  
}
```

```
.neon-nav-right {  
    margin-right: 20px;  
    gap: 15px;  
}  
  
}
```

SKILL MATRIX

```
.neon-nav-left li,  
.neon-nav-right li {  
    list-style: none;  
}  
  
.neon-nav-left a,  
.neon-nav-right a {  
    text-decoration: none;  
    color: #0ef;  
    padding: 10px 20px;  
    border: 2px solid #0ef;  
    border-radius: 5px;  
    transition: all 0.3s ease;  
    font-weight: 600;  
    text-transform: uppercase;  
    font-size: 0.9rem;  
}  
  
.neon-nav-left a:hover,  
.neon-nav-right a:hover {  
    background-color: #0ef;  
    color: #1a1a1a;  
    box-shadow: 0 0 10px #0ef, 0 0 20px #0ef, 0 0 30px #0ef;
```

SKILL MATRIX

}

```
.content {  
    padding: 100px 20px;  
    max-width: 1200px;  
    width: 100%;  
}
```

```
.section {  
    margin-bottom: 40px;  
    background-color: #fff;  
    padding: 20px;  
    border-radius: 10px;  
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);  
    transition: box-shadow 0.3s ease;  
}
```

```
.section:hover {  
    box-shadow: 0 8px 16px rgba(0, 0, 0, 0.2);  
}
```

```
.section h2 {  
    font-size: 2em;  
    margin-bottom: 20px;
```

SKILL MATRIX

```
text-align: center;  
color: #0099ff;  
}  
  
}
```

```
.section ul {  
display: grid;  
grid-template-columns: repeat(3, 1fr);  
gap: 20px;  
list-style-type: none;  
padding: 0;  
width: 100%;  
}  
  
}
```

```
.section li {  
background-color: transparent;  
padding: 0;  
border: 2px solid #0099ff;  
border-radius: 10px;  
text-align: center;  
transition: background-color 0.3s ease, transform 0.3s ease;  
cursor: pointer;  
}  
  
}
```

```
.section li:hover {
```

SKILL MATRIX

```
background-color: #0099ff;  
color: #fff;  
transform: translateY(-5px);  
}
```

```
.section li a {  
display: block;  
height: 100%;  
width: 100%;  
color: inherit;  
text-decoration: none;  
font-size: 1.2em;  
font-weight: bold;  
padding: 20px;  
box-sizing: border-box;  
}
```

```
footer {  
background-color: #1a1a1a;  
color: #fff;  
padding: 20px;  
text-align: center;  
margin-top: -100px;  
width: 100%;
```

SKILL MATRIX

```
}

</style>

</head>

<body>

<nav class="neon-navbar">

<ul class="neon-nav-left">

<li><a href="AdminHome.jsp">Home</a></li>

<li><a href="ChangePasswordAdmin.jsp">Change Password</a></li>

<li><a href="index.html">Logout</a></li>

</ul>

<ul class="neon-nav-right">

<!-- You can add additional links here if required -->

</ul>

</nav>

<div class="content">

<section class="section">

<h2>Admin Module</h2>

<ul>

<li><a href="view_student_info.jsp">View Student Information</a></li>

<li><a href="search_student_info.jsp">Search Student Information</a></li>

<li><a href="add_question.jsp">Add Questions</a></li>

<li><a href="update_question.jsp">Update Questions</a></li>


```

SKILL MATRIX

```
<li><a href="delete_question.jsp">Delete Questions</a></li>
<li><a href="view_question_list.jsp">View Question List</a></li>
<li><a href="view_messages.jsp">View Messages</a></li>
</ul>
</section>
</div>

<footer>
<p>&copy; 2024 Admin Module. All rights reserved.</p>
</footer>

</body>
</html>
```

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Student Module</title>
<style>
```

SKILL MATRIX

```
@import
url('https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;600;700&display=swap'
);

body {
    font-family: 'Poppins', sans-serif;
    margin: 0;
    padding: 0;
    display: flex;
    flex-direction: column;
    align-items: center;
    background-color: #f4f4f4;
    color: #333;
}

.neon-navbar {
    display: flex;
    justify-content: space-between;
    align-items: center;
    padding: 15px 30px;
    background-color: #1a1a1a;
    box-shadow: 0 2px 10px rgba(0, 78, 146, 0.7);
    position: sticky;
    top: 0;
    width: 100%;
}
```

SKILL MATRIX

```
z-index: 1000;
```

```
}
```

```
.neon-nav-left,
```

```
.neon-nav-right {
```

```
    display: flex;
```

```
    gap: 25px;
```

```
    align-items: center;
```

```
}
```

```
.neon-nav-right {
```

```
    margin-right: 20px;
```

```
    gap: 15px;
```

```
}
```

```
.neon-nav-left li,
```

```
.neon-nav-right li {
```

```
    list-style: none;
```

```
}
```

```
.neon-nav-left a,
```

```
.neon-nav-right a {
```

```
    text-decoration: none;
```

```
    color: #0ef;
```

SKILL MATRIX

```
padding: 10px 20px;  
border: 2px solid #0ef;  
border-radius: 5px;  
transition: all 0.3s ease;  
font-weight: 600;  
text-transform: uppercase;  
font-size: 0.9rem;  
}
```

```
.neon-nav-left a:hover,  
.neon-nav-right a:hover {  
background-color: #0ef;  
color: #1a1a1a;  
box-shadow: 0 0 10px #0ef, 0 0 20px #0ef, 0 0 30px #0ef;  
}
```

```
.content {  
padding: 100px 20px;  
max-width: 1200px;  
width: 100%;  
}
```

```
.section {  
margin-bottom: 40px;
```

SKILL MATRIX

```
background-color: #fff;  
padding: 20px;  
border-radius: 10px;  
box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);  
transition: box-shadow 0.3s ease;  
}  
  
.
```

```
.section:hover {  
    box-shadow: 0 8px 16px rgba(0, 0, 0, 0.2);  
}
}
```

```
.section h2 {  
    font-size: 2em;  
    margin-bottom: 20px;  
    text-align: center;  
    color: #0099ff;  
}
}
```

```
.section ul {  
    display: grid;  
    grid-template-columns: repeat(3, 1fr);  
    gap: 20px;  
    list-style-type: none;  
    padding: 0;
}
```

SKILL MATRIX

```
width: 100%;  
}  
  
.section li {  
background-color: transparent;  
padding: 0;  
border: 2px solid #0099ff;  
border-radius: 10px;  
text-align: center;  
transition: background-color 0.3s ease, transform 0.3s ease;  
cursor: pointer;  
}  
  
.section li:hover {  
background-color: #0099ff;  
color: #fff;  
transform: translateY(-5px);  
}  
  
.section li a {  
display: block;  
height: 100%;  
width: 100%;  
color: inherit;
```

SKILL MATRIX

```
text-decoration: none;  
font-size: 1.2em;  
font-weight: bold;  
padding: 20px;  
box-sizing: border-box;  
}  
  
footer {
```

```
background-color: #1a1a1a;  
color: #fff;  
padding: 20px;  
text-align: center;  
margin-top: -50PX;  
width: 100%;  
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<nav class="neon-navbar">  
  <ul class="neon-nav-left">  
    <li><a href="StudentHome.jsp">Home</a></li>  
    <li><a href="ChangePasswordStudent.jsp">Change Password</a></li>  
    <li><a href="index.html">Logout</a></li>
```

SKILL MATRIX

```
</ul>

<ul class="neon-nav-right">
    <!-- You can add additional links here if required --&gt;
&lt;/ul&gt;

&lt;/nav&gt;

&lt;div class="content"&gt;
    &lt;section class="section"&gt;
        &lt;h2&gt;Student Module&lt;/h2&gt;
        &lt;ul&gt;
            &lt;li&gt;&lt;a href="ViewProfile.jsp"&gt;Profile&lt;/a&gt;&lt;/li&gt;
            &lt;li&gt;&lt;a href="EditProfile.jsp"&gt;Edit Profile&lt;/a&gt;&lt;/li&gt;
            &lt;li&gt;&lt;a href="Exam.jsp"&gt;Start Exam&lt;/a&gt;&lt;/li&gt;
            &lt;li&gt;&lt;a href="ViewResult.jsp"&gt;View Result&lt;/a&gt;&lt;/li&gt;
            &lt;li&gt;&lt;a href="DeleteAccount.jsp"&gt;Delete Your Account&lt;/a&gt;&lt;/li&gt;
        &lt;/ul&gt;
    &lt;/section&gt;
&lt;/div&gt;

&lt;footer&gt;
    &lt;p&gt;&amp;copy; 2024 Student Module. All rights reserved.&lt;/p&gt;
&lt;/footer&gt;

&lt;/body&gt;</pre>
```

SKILL MATRIX

```
</html>

package DB;

import java.sql.*;

public class DBCon

{

    public Connection con;

    public Statement stmt;

    public PreparedStatement pstmt;

    public ResultSet rst;

    public DBCon()

    {

        try

        {

            Class.forName("com.mysql.jdbc.Driver");

            con=DriverManager.getConnection("jdbc:mysql://localhost:3306/oes","root","root");

        }

        catch(Exception e)

        {

            e.printStackTrace();

        }

    }

}
```

SKILL MATRIX

```
package admin;

import DB.DBCon;

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

public class AChangePassword extends HttpServlet {

    protected void service(HttpServletRequest request, HttpServletResponse response)

        throws ServletException, IOException {

        response.setContentType("text/html;charset=UTF-8");

        try (PrintWriter out = response.getWriter()) {

            HttpSession session=request.getSession();

            String user=session.getAttribute("user").toString();

            String opswd=request.getParameter("oldPassword");

            String npswd=request.getParameter("newPassword");

            String cpswd=request.getParameter("confirmPassword");

            DBCon db=new DBCon();

            db.pstmt=db.con.prepareStatement("select * from login where u_name=? and

pswd=?");

            db.pstmt.setString(1, user);

            db.pstmt.setString(2, opswd);

            db.rst=db.pstmt.executeQuery();
```

SKILL MATRIX

```
if(db.rst.next())
{
    if(npswd.equals(cpswd))
    {
        db.pstmt=db.con.prepareStatement("update login set pswd=? where u_name=?");
        db.pstmt.setString(1, npswd);
        db.pstmt.setString(2, user);
        int i=db.pstmt.executeUpdate();
        if(i>0)
        {
            response.sendRedirect("ChangePasswordAdmin.jsp?success=Password Updated
Successfully");
        }
    }
    else
    {
        response.sendRedirect("ChangePasswordAdmin.jsp?error>New and Confirm
Password must be Same");
    }
}
else
{
    response.sendRedirect("ChangePasswordAdmin.jsp?error=Old Password does not
Correct");
}
```

SKILL MATRIX

```
}

catch(Exception e)

{

    e.printStackTrace();

}

}

}

package admin;

import DB.DBCon;

import java.io.IOException;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

public class AddQuestion extends HttpServlet {

    protected void service(HttpServletRequest request, HttpServletResponse response)

        throws ServletException, IOException {

        response.setContentType("text/html;charset=UTF-8");

        try {

            String ques_id = request.getParameter("ques_id");

            String question = request.getParameter("question");

            String option1 = request.getParameter("option1");

            String option2 = request.getParameter("option2");


```

SKILL MATRIX

```
String option3 = request.getParameter("option3");

String option4 = request.getParameter("option4");

String correctAnswer = request.getParameter("correctAnswer");

DBCon db = new DBCon();

db.pstmt = db.con.prepareStatement("INSERT INTO ques_mstr (ques_id, que, op1,
op2, op3, op4, ans) VALUES (?, ?, ?, ?, ?, ?, ?, ?)");

db.pstmt.setString(1, ques_id);

db.pstmt.setString(2, question);

db.pstmt.setString(3, option1);

db.pstmt.setString(4, option2);

db.pstmt.setString(5, option3);

db.pstmt.setString(6, option4);

db.pstmt.setString(7, correctAnswer);

int rowsInserted = db.pstmt.executeUpdate();

if (rowsInserted > 0) {

    response.sendRedirect("add_question.jsp?success=Question added successfully with
Question ID: " + ques_id);

} else {

    response.sendRedirect("add_question.jsp?error=Failed to add question. Try again.");

}

} catch (Exception e) {

    e.printStackTrace();

}

}
```

SKILL MATRIX

```
package admin;

import DB.DBCon;

import java.io.IOException;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

public class DeleteQuestion extends HttpServlet {

    protected void service(HttpServletRequest request, HttpServletResponse response)

        throws ServletException, IOException {

        response.setContentType("text/html;charset=UTF-8");

        try {

            String ques_id = request.getParameter("questionId");

            DBCon db = new DBCon();

            db.pstmt = db.con.prepareStatement("DELETE FROM ques_mstr WHERE ques_id = ?");

            db.pstmt.setString(1, ques_id);

            int rowsDeleted = db.pstmt.executeUpdate();

            if (rowsDeleted > 0) {

                response.sendRedirect("delete_question.jsp?success=Question deleted successfully with Question ID: " + ques_id);

            } else {

                response.sendRedirect("delete_question.jsp?error=Failed to delete the question. Please try again.");

            }

        }

    }

}
```

SKILL MATRIX

```
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
  
package admin;  
  
import java.io.IOException;  
  
import java.io.PrintWriter;  
  
import javax.servlet.ServletException;  
  
import javax.servlet.http.HttpServlet;  
  
import javax.servlet.http.HttpServletRequest;  
  
import javax.servlet.http.HttpServletResponse;  
  
import javax.servlet.http.HttpSession;  
  
public class LoginValidate extends HttpServlet {  
  
    protected void service(HttpServletRequest request, HttpServletResponse response)  
        throws ServletException, IOException {  
        response.setContentType("text/html;charset=UTF-8");  
        try (PrintWriter out = response.getWriter())  
        {  
            String user = request.getParameter("t2");  
            String pswd = request.getParameter("t3");  
            DB.DBCon db = new DB.DBCon();  
            db.pstmt = db.con.prepareStatement("select * from login where u_name=? and  
            pswd=?");  
        }  
    }  
}
```

SKILL MATRIX

```
db.pstmt.setString(1, user);

db.pstmt.setString(2, pswd);

db.rst = db.pstmt.executeQuery();

if (db.rst.next())

{

    String utype = db.rst.getString("type");

    HttpSession session = request.getSession();

    session.setAttribute("user", user);

    session.setAttribute("role", utype);

    if (utype.equalsIgnoreCase("Admin"))

    {

        response.sendRedirect("AdminHome.jsp");

    }

    else if (utype.equalsIgnoreCase("Student"))

    {

        response.sendRedirect("StudentHome.jsp");

    }

}

else

{

    response.sendRedirect("Login.jsp?error=Incorrect Username or Password");

}

}

catch (Exception e)
```

SKILL MATRIX

```
{  
    e.printStackTrace();  
}  
}  
}  
  
package admin;  
  
import DB.DBCon;  
  
import java.io.IOException;  
  
import javax.servlet.ServletException;  
  
import javax.servlet.http.HttpServlet;  
  
import javax.servlet.http.HttpServletRequest;  
  
import javax.servlet.http.HttpServletResponse;  
  
public class UpdateQuestion extends HttpServlet {  
  
    protected void service(HttpServletRequest request, HttpServletResponse response)  
        throws ServletException, IOException {  
  
        response.setContentType("text/html;charset=UTF-8");  
  
        try {  
  
            String ques_id = request.getParameter("ques_id");  
  
            String question = request.getParameter("question");  
  
            String option1 = request.getParameter("option1");  
  
            String option2 = request.getParameter("option2");  
  
            String option3 = request.getParameter("option3");  
  
            String option4 = request.getParameter("option4");  
        }
```

SKILL MATRIX

```
String correctAnswer = request.getParameter("correctAnswer");

DBCon db = new DBCon();

db.pstmt = db.con.prepareStatement("UPDATE ques_mstr SET que = ?, op1 = ?, op2 =
?, op3 = ?, op4 = ?, ans = ? WHERE ques_id = ?");

db.pstmt.setString(1, question);

db.pstmt.setString(2, option1);

db.pstmt.setString(3, option2);

db.pstmt.setString(4, option3);

db.pstmt.setString(5, option4);

db.pstmt.setString(6, correctAnswer);

db.pstmt.setString(7, ques_id);

int rowsUpdated = db.pstmt.executeUpdate();

if (rowsUpdated > 0) {

    response.sendRedirect("update_question.jsp?success=Question updated successfully
with Question ID: " + ques_id);

} else {

    response.sendRedirect("update_question.jsp?error=Failed to update the question.
Please try again.");

}

} catch (Exception e) {

    e.printStackTrace();

}

}
```

SKILL MATRIX

```
package student;

import DB.DBCon;

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

public class DeleteAccount extends HttpServlet {

    protected void service(HttpServletRequest request, HttpServletResponse response)

        throws ServletException, IOException {

        response.setContentType("text/html;charset=UTF-8");

        HttpSession session = request.getSession(false);

        try (PrintWriter out = response.getWriter()) {

            if (session != null) {

                String username = (String) session.getAttribute("user");

                DBCon db = new DBCon();

                db.con.setAutoCommit(false);

                db.pstmt = db.con.prepareStatement("DELETE FROM login WHERE u_name = ?");

                db.pstmt.setString(1, username);

                int loginDeleted = db.pstmt.executeUpdate();

                db.pstmt = db.con.prepareStatement("DELETE FROM result_mstr WHERE user_id

= ?");

                db.pstmt.setString(1, username);

```

SKILL MATRIX

```
int resultDeleted = db.pstmt.executeUpdate();

out.println("<h2>Debug Info:</h2>");

out.println("<p>Login deleted rows: " + loginDeleted + "</p>");

out.println("<p>Result deleted rows: " + resultDeleted + "</p>");

if (loginDeleted > 0 || resultDeleted > 0) {

    db.con.commit();

    session.invalidate();

    response.sendRedirect("index.html");

} else {

    db.con.rollback();

    out.println("<h2>Failed to delete the account. No records found to delete.</h2>");

}

db.con.close();

} else {

    out.println("<h2>You are not logged in. Please log in first.</h2>");

}

} catch (Exception e) {

    e.printStackTrace();

}

}
```

SKILL MATRIX

```
package student;

import DB.DBCon;

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

public class DeleteAccount extends HttpServlet {

    protected void service(HttpServletRequest request, HttpServletResponse response)

        throws ServletException, IOException {

        response.setContentType("text/html;charset=UTF-8");

        HttpSession session = request.getSession(false);

        try (PrintWriter out = response.getWriter()) {

            if (session != null) {

                String username = (String) session.getAttribute("user");

                DBCon db = new DBCon();

                db.con.setAutoCommit(false);

                db.pstmt = db.con.prepareStatement("DELETE FROM login WHERE u_name = ?");

                db.pstmt.setString(1, username);

                int loginDeleted = db.pstmt.executeUpdate();

                db.pstmt = db.con.prepareStatement("DELETE FROM result_mstr WHERE user_id

= ?");

                db.pstmt.setString(1, username);

```

SKILL MATRIX

```
int resultDeleted = db.pstmt.executeUpdate();

out.println("<h2>Debug Info:</h2>");

out.println("<p>Login deleted rows: " + loginDeleted + "</p>");

out.println("<p>Result deleted rows: " + resultDeleted + "</p>");

if (loginDeleted > 0 || resultDeleted > 0) {

    db.con.commit();

    session.invalidate();

    response.sendRedirect("index.html");

} else {

    db.con.rollback();

    out.println("<h2>Failed to delete the account. No records found to delete.</h2>");

}

db.con.close();

} else {

    out.println("<h2>You are not logged in. Please log in first.</h2>");

}

} catch (Exception e) {

    e.printStackTrace();

}

}
```

SKILL MATRIX

```
package student;

import DB.DBCon;

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

public class SChangePassword extends HttpServlet {

    protected void service(HttpServletRequest request, HttpServletResponse response)

        throws ServletException, IOException {

        response.setContentType("text/html;charset=UTF-8");

        try (PrintWriter out = response.getWriter()) {

            HttpSession session=request.getSession();

            String user=session.getAttribute("user").toString();

            String opswd=request.getParameter("oldPassword");

            String npswd=request.getParameter("newPassword");

            String cpswd=request.getParameter("confirmPassword");

            DBCon db=new DBCon();

            db.pstmt=db.con.prepareStatement("select * from login where u_name=? and

pswd=?");

            db.pstmt.setString(1, user);

            db.pstmt.setString(2, opswd);

            db.rst=db.pstmt.executeQuery();
```

SKILL MATRIX

```
if(db.rst.next())  
  
{  
  
    if(npswd.equals(cpswd))  
  
    {  
  
        db.pstmt=db.con.prepareStatement("update login set pswd=? where u_name=?");  
  
        db.pstmt.setString(1, npswd);  
  
        db.pstmt.setString(2, user);  
  
        int i=db.pstmt.executeUpdate();  
  
        if(i>0)  
  
        {  
  
            response.sendRedirect("ChangePasswordStudent.jsp?success=Password  
Updated Successfully");  
  
        }  
  
    }  
  
    else  
  
    {  
  
        response.sendRedirect("ChangePasswordStudent.jsp?error>New and Confirm  
Password must be Same");  
  
    }  
  
}  
  
else  
  
{  
  
    response.sendRedirect("ChangePasswordStudent.jsp?error=Old Password does not  
Correct");  
  
}
```

SKILL MATRIX

```
        }

    catch(Exception e)

    {

        e.printStackTrace();

    }

}

package student;

import DB.DBCon;

import java.io.IOException;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

public class UpdateProfile extends HttpServlet {

    protected void service(HttpServletRequest request, HttpServletResponse response)

        throws ServletException, IOException {

        response.setContentType("text/html;charset=UTF-8");

        try {

            HttpSession session=request.getSession();

            String username = session.getAttribute("user").toString();

            String dob = request.getParameter("dob");


```

SKILL MATRIX

```
String gender = request.getParameter("gender");

String quali1 = request.getParameter("quali1");

String quali2 = request.getParameter("quali2");

String address = request.getParameter("addr");

String city = request.getParameter("city");

String contact = request.getParameter("contact");

String email = request.getParameter("email");

DBCon db = new DBCon();

db.pstmt = db.con.prepareStatement("UPDATE user_info SET dob = ?, gender = ?,
quali1 = ?, quali2 = ?, addr = ?, city = ?, contact = ?, email = ? WHERE u_name = ?");

db.pstmt.setString(1, dob);

db.pstmt.setString(2, gender);

db.pstmt.setString(3, quali1);

db.pstmt.setString(4, quali2);

db.pstmt.setString(5, address);

db.pstmt.setString(6, city);

db.pstmt.setString(7, contact);

db.pstmt.setString(8, email);

db.pstmt.setString(9, username);

int i = db.pstmt.executeUpdate();

if (i > 0) {

    response.sendRedirect("EditProfile.jsp?success=Profile updated successfully");

} else {

    response.sendRedirect("EditProfile.jsp?error=Failed to update the profile. Please try
again.");
}
```

SKILL MATRIX

```
    }  
  
} catch (Exception e) {  
  
    e.printStackTrace();  
  
}  
  
}  
  
}
```

SKILL MATRIX

Chapter 5: Result and Conclusion

This section discusses the result of work done in this project and also mentions the future scope improvement.

Results

- 1. Successful Implementation:** The system has been successfully implemented, meeting the objectives outlined in the planning phase. The key features and functionalities have been thoroughly tested, ensuring robust performance.
- 2. Enhanced Modularity:** By adopting modular design principles, the project achieves a structured and organized architecture. This reduces complexity and simplifies maintenance.
- 3. Improved User Experience:**
 - For administrators, managing users, adding or deleting questions, and posting news has become seamless.
 - For users, taking exams, viewing results, and sharing feedback are now streamlined and user-friendly.
- 4. System Adaptability:** The application is portable, allowing it to operate on various computer systems with different operating systems and standards. This flexibility ensures wider adoption and usability.
- 5. Data Accuracy and Security:** The system ensures proper handling of user inputs and terminates operations gracefully, avoiding crashes or unexpected behaviour.

SKILL MATRIX

Conclusion

The software will be developed by implementing the concept of modularity which in turn reduces the complexity involved in maintaining it. The administrator should have a sound technical knowledge about maintaining the software and further enhancements will be undertaken by the developer.

The application is portable which ensure its adaptability for use on different computer terminals with different operating system and standards.

The factors guarantee the software's availability includes proper termination and correct input details.

Hence, we may conclude that the application system being developed helps a great deal in modifying the computerized **SKILL MATRIX SYSTEM**.

Future Scope

Although the Skill Matrix System addresses current needs effectively, the following enhancements can further improve its functionality and relevance:

1. Integration with Advanced Technologies:

- Artificial Intelligence (AI): For personalized learning and adaptive question patterns.
- Machine Learning (ML): To analyze user performance trends and provide insights.
- Cloud Computing: For seamless data storage and remote access.

2. Enhanced User Experience:

- Developing a mobile-friendly version of the application.
- Incorporating gamification features to make exams more engaging.

3. Multi-Language Support:

- Adding support for multiple languages to cater to diverse user demographics.

SKILL MATRIX

4. Advanced Security Features:

- Implementing biometric or multi-factor authentication for enhanced security.

5. Analytics Dashboard:

- Providing administrators with a detailed dashboard for tracking user activity, performance metrics, and system usage.

6. Offline Access:

- Incorporating offline functionality to ensure accessibility in areas with limited internet connectivity.

The **Skill Matrix System** has been successfully developed and implemented, proving its value in simplifying and automating key processes. Its modularity and adaptability ensure long-term usability and relevance. With a roadmap for future improvements, the system is well-positioned to evolve and cater to the changing needs of users and administrators.

SKILL MATRIX

Bibliography

Reference Books:

- Pankaj Sharma, **Introduction to Web Technology**, 2nd Edition, S.K. Kataria & Sons Publication, April 2007
- Shrika Jain, **Information System**, 3rd Edition, Pregti Publication, April 2004
- Roger S. Pressman, **Software Engineering**, 3rd Edition, McGraw Hill International, April 2002
- Herbert Schildt, **Java 2: Complete Reference**, 5th Edition, Pregti Publication, April 2004

Reference Websites:

- www.stardeveloper.com
- www.javaskyline.com
- www.javaworld.com