

PROJECT BASED LEARNING (PBL-1) REPORT

DESKTOP BASED AUDITING APPLICATION



Submitted by:

| S.No. | Name | PRN |
| --- | --- | --- |
| 1 | Jay Gondaliya | 22070122088 |
| 2 | K.Rishik Kumar Reddy | 22070122091 |
| 3 | Dhairya Mehra | 22070122118 |
| 4 | Rutu Bhanderi | 22070122168 |

Under the guidance of:

**Dr. Kanhaiya Sharma**

**Department of Computer Science**

**SYMBIOSIS INSTITUTE OF TECHNOLOGY, PUNE**

**Introduction**

SIT Pune requires a robust auditing system to streamline its auditing processes, enhance data management, and improve overall efficiency. The existing manual methods are time-consuming, prone to errors, and lack comprehensive reporting capabilities. To address these challenges, SIT Pune aims to develop a desktop-based auditing application that integrates seamlessly with its operations.

**Objectives**

The primary objective is to develop a feature-rich desktop application using Java AWT for the user interface and MySQL for data storage, specifically tailored to meet the auditing requirements of SIT Pune. The application should provide functionalities for recording auditing data, generating reports, managing user accounts, and ensuring data security.

**Current system**

Professors at the institution presently engage in a manual process for completing mid-semester and end-semester audit forms, wherein they are required to fill out these forms by hand on paper documents. This method proves to be inefficient as it consumes a significant amount of time and is prone to errors arising from manual transcription.

**Drawbacks**

1. **Human Error:** Manual auditing is prone to human error, such as oversight, misinterpretation of data, or data entry mistakes.

2. **Time-consuming:** Manual auditing processes are typically slower than automated systems, requiring more time to collect, analyze, and verify data.

3. **Resource Intensive:** Manual audits often require a significant allocation of human resources, including time, personnel, and expertise, which can increase costs.

4. **Security Risks:** Manual handling of sensitive information increases the risk of data breaches or unauthorized access compared to automated systems with built-in security measures.

5. **Difficulty in Documentation:** Maintaining comprehensive documentation of manual audit processes can be cumbersome, leading to gaps or inconsistencies in record-keeping

**Features implemented in our application**

* **User Authentication:** Implement login functionality to authenticate users.
* **Data Entry Forms:** Create forms to input auditing data.
* **Data Validation:** Validate user input to ensure data integrity.
* **Report Generation:** Allow users to generate reports based on the auditing data.
* **Data Export:** Allow users to export auditing data in different formats.
* **Search and Filter:** Implement functionality to search and filter auditing data.
* **User Management:** Allow administrators to manage user accounts and permissions.

**Software and Hardware requirements**

1. **Software:**

* Java AWT for the desktop user interface.
* MySQL for database management.
* JDBC for database connectivity.
* Security mechanisms to protect sensitive data.
* Proper exception handling and error reporting for robustness.
* Appropriate version of the Java Runtime Environment installed to run Java applications.
* Can run on any operating system such as windows , linux and mac os.

1. **Hardware:**

* Multi core processor with at least 2 cores.
* Minimum 4 GB RAM , for larger applications 8 GB RAM.
* Storage 5 GB.
* Stable network connection.
* GPU is optional.
* Keyboard , mouse, monitor for viewing and editing.

**Timeline**

* **Week 1-2:** Requirements Gathering and Design

1. Gather initial requirements from stakeholders and conduct analysis.
2. Refine requirements and finalize the system architecture and database schema design.

* **Week 3-4**: Database Setup and Backend Development

1. Set up a MySQL database environment and create necessary tables.
2. Begin backend development, implementing core functionalities such as user management and auditing logic.

* **Week 5-6:** Frontend Development

1. Design the user interface using Java AWT, focusing on basic layouts and components.
2. Continue frontend development, implementing key UI features and navigation.

* **Week 7-8:** Integration and Testing

1. Integrate backend and frontend components to create a functional application.
2. Conduct thorough testing, including unit testing, integration testing, and system testing.

* **Week 9-10:** Refinement and Finalization

1. Implement any required refinements based on user feedback and testing results.
2. Finalize documentation and project deliverables.

* **Week 11-12:** Final Testing and Project Completion

1. Conduct final testing and quality assurance checks to ensure the application meets requirements.
2. Address any last-minute adjustments or issues.
3. Project handover to stakeholders and project closure.