

A Project Report
on
BVRITH ECAP REPLICA WEBSITE

submitted in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

by

22WH1A05D5

22WH1A05D2

22WH1A05F0

22WH1A05I7

22WH1A05G4

G. VEDASRI LAKSHMI

G. SAI SARASWATHI

B.BHAVANA

T.SIRI CHANDANA

K.JOSHITHA

under the esteemed guidance of

Dr. VENKATESH B

Associate Professor



Department of Computer Science and Engineering

BVRIT HYDERABAD COLLEGE OF ENGINEERING FOR WOMEN

(NBA Accredited – EEE, ECE, CSE,AIML and IT)

(Approved by AICTE, New Delhi and Affiliated to JNTUH, Hyderabad)

Bachupally, Hyderabad – 500090

July, 2024

DECLARATION

We hereby declare that the work presented in this project entitled “**BVRITH ECAP Replica Website**” submitted towards the completion of Project Work in the II year of B.Tech., CSE at BVRIT Hyderabad College of Engineering for Women, Hyderabad, is an authentic record of our original work carried out under the guidance of **Dr. Venkatesh B**, Associate Professor, Department of CSE.

G.VEDASRI LAKSHMI

(22WH1A05D5)

G.SAI SARASWATHI

(22WH1A05D2)

B.BHAVANA

(22WH1A05F0)

T.SIRI CHANDANA

(22WH1A05I7)

K.JOSHITHA

(22WH1A05)

BVRIT HYDERABAD COLLEGE OF ENGINEERING FOR WOMEN
(NBA Accredited – EEE, ECE, CSE,AIML and IT)
(Approved by AICTE, New Delhi and Affiliated to JNTUH, Hyderabad)

Bachupally, Hyderabad – 500090

Department of Computer Science and Engineering



CERTIFICATE

This is to certify that the Project Work report on “**BVRITH ECAP Replica Website**” is a bonafide work carried out by Ms. G Veda Sri Lakshmi (22WH1A05D5), Ms. G Saraswathi (22WH1A05D2), Ms. B Bhavana (22WH1A05F0), Ms. T Siri Chandana (22WH1A05I7), and Ms. K Joshitha (22WH1A05G4) in partial fulfillment of the award of the B.Tech. degree in **Computer Science and Engineering, BVRIT Hyderabad College of Engineering for Women, Bachupally, Hyderabad**, affiliated to Jawaharlal Nehru Technological University Hyderabad, Hyderabad under my guidance and supervision. The results embodied in the project work have not been submitted to any other University or Institute for the award of any degree or diploma.

Head of the Department

Dr. M Sree Vani,
Professor,
Department of CSE

Internal Guide

Dr. B Venkatesh
Associative Professor
Department Professor

Contents

ABSTRACT

The BVRITH ECAP (Engineering College Automation Platform) replica website represents a pivotal advancement in the automation and optimization of administrative, academic, and communication processes within an engineering college setting. This documentation presents a detailed chronicle of the website's developmental journey, offering extensive insights into its meticulous design, robust implementation, and prospective avenues for future enhancement. The primary objective of the platform is to streamline and enhance the management of critical institutional facets, including student records, faculty information, course particulars, and comprehensive institutional data. By leveraging cutting-edge technologies and intuitive design principles, the website ensures a cohesive and intuitive user experience for all stakeholders involved in the educational ecosystem. Embodying a commitment to efficiency and user-centricity, the BVRITH ECAP replica website stands as a testament to innovation in educational administration, promising sustained operational excellence and adaptability in the face of evolving educational landscapes.

TABLE OF CONTENTS

ABSTRACT	i
LIST OF FIGURES	iv
1 INTRODUCTION	
1.1 Objectives	1
1.2 Methodology	2
2 DESIGN	
2.1 Introduction	3
2.2 Architecture Diagram	4
2.3 E-R Diagram	4
2.4 Use-Case Diagram	8
3 TECHNOLOGY STACK	
3.1 Introduction	9
3.2 Software Requirements	9
3.3 Hardware Requirements	11
3.4 Packages and Modules	12

4 IMPLEMENTATION

4.1 Introduction	14
4.2 Test-Cases	14
4.3 Testing	18

5 CONCLUSION AND FUTURE SCOPE

5.1 Conclusion	23
5.2 Future Scope	24

6 REFERENCE

25

LIST OF FIGURES

3.1 Architecture Diagram	4
3.2 E-R Diagram	6
3.3 Sequential Diagram	8
3.4 Testing	22

1 INTRODUCTION

Introduction

In response to the evolving demands of modern educational institutions, the BVRITH ECAP replica website stands as a comprehensive solution aimed at centralizing and optimizing various facets of college administration. This platform integrates essential functionalities such as student and faculty management, course administration, academic scheduling, and robust communication tools. By leveraging technological advancements, the website enhances operational efficiency and facilitates seamless interactions within the college community.

1.1 Objectives

The primary objectives of the BVRITH ECAP replica website are as follows:

1. **Centralized Administration:** Provide a unified platform for managing diverse administrative tasks, including student enrolment, attendance tracking, and fee management, to streamline operations and reduce administrative overhead.
2. **Enhanced Academic Management:** Facilitate effective management of academic processes, including course scheduling, curriculum planning, grading, and assessment, to ensure smooth academic operations and adherence to educational standards.
3. **Improved Communication:** Foster better communication channels among students, faculty, and administrative staff through integrated messaging systems, announcements, and notifications, promoting transparency and timely dissemination of information.
4. **User-Centric Design:** Develop a user-friendly interface that caters to the distinct needs of students, faculty members, and administrators, ensuring ease of use, accessibility, and efficiency in navigating the platform.
5. **Data Security and Privacy:** Implement robust security measures, including secure login mechanisms, data encryption, and access control, to safeguard sensitive information and maintain confidentiality.

6. **Scalability and Flexibility:** Design a scalable architecture capable of accommodating future growth and technological advancements, allowing for seamless integration of new features and functionalities

1.2 Methodology

The methodology for collecting ECAP information at BVRITH involves a structured approach combining surveys and focus group discussions. First, we distribute comprehensive surveys to all students to gather data on enrollment status, career aspirations, and related demographic details. Following this, we conduct focus group discussions with selected students to delve deeper into their career goals and the factors influencing their decisions. The collected data is then analyzed using statistical methods for quantitative responses and thematic analysis for qualitative insights, providing a thorough understanding of student aspirations and enrollment patterns at BVRITH.

2 DESIGN

2.1 Introduction

The design phase of the BVRITH ECAP replica website involved a comprehensive approach to meet user requirements and operational objectives. Key design principles encompassed:

- **User Interface (UI):** Creating an intuitive and responsive interface tailored to diverse user roles, ensuring ease of navigation and functionality accessibility.
- **Database Design:** Developing a robust and scalable database architecture to efficiently manage large volumes of data related to students, courses, faculty, and administrative records.
- **Security Measures:** Implementing stringent security protocols to protect sensitive data from unauthorized access and cyber threats, ensuring compliance with data privacy regulations.

2.2 Architecture Diagram

The architecture diagram illustrates the structural layout of the BVRITH ECAP replica website, showcasing the integration of various components and their interactions. Below is a simplified Architecture representation as follows:

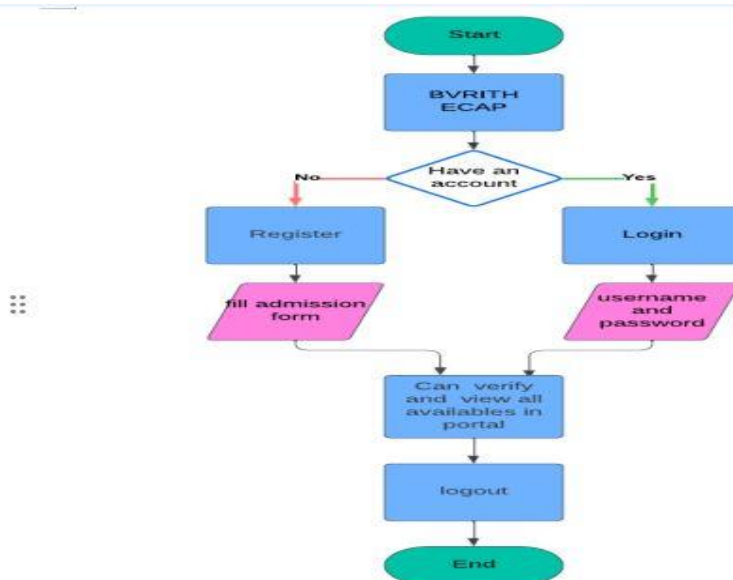


Fig. 2.2 flow chart

To access the **BVRITH ECAP** website, start by logging in with your user ID and password if you already have an account. Once logged in, you'll be taken to the menu page, where you can access various features like checking your attendance, viewing assignments, and making online fee payments. Additionally, you can view your bio-data and use the "Edit Profile" feature to update any incorrect details. After you've finished using the portal, you can log out and close the website.

If you're new to the **BVRITH ECAP** portal or don't have an account, you need to register first. During registration, you'll fill out the admission form with your academic details, phone number, and other necessary information. Once the form is completed, you can log in to the website and access all your academic details through the portal. When you're done, you can log out and close the website.

2.3 E-R DIAGRAM

An E-R diagram, or Entity-Relationship diagram, is a visual representation of the relationships between different data entities in a database. It helps to illustrate how data is structured and how different entities (like customers, orders, products) are related to each other.

In simple terms, it's like a map that shows how different pieces of data connect and interact within a system.

An Entity-Relationship (E-R) diagram is a tool used in database design to visually depict the structure of a database. It consists of entities, relationships, and attributes:

1. **Entities:** These are objects or things in the real world that have distinct existence and are represented by rectangles. For example, in a university database, entities could be "Student," "Course," and "Professor."
2. **Attributes:** These are properties or characteristics of entities, represented by ovals. For instance, the "Student" entity might have attributes like "StudentID," "Name," and "DateOfBirth."
3. **Relationships:** These illustrate how entities are related to each other and are represented by diamonds. For example, a "Student" entity might have a relationship with a "Course" entity, which could be "enrolls in."
4. **Cardinality:** This shows the numerical relationship between entities. For example, a single student can enroll in multiple courses (one-to-many relationship), or a course can have many students enrolled in it (many-to-one relationship).
5. **Entities:** These are objects or things in the real world that have distinct existence and are represented by rectangles. For example, in a university database, entities could be "Student," "Course," and "Professor."
6. **Attributes:** These are properties or characteristics of entities, represented by ovals. For instance, the "Student" entity might have attributes like "StudentID," "Name," and "DateOfBirth."
7. **Relationships:** These illustrate how entities are related to each other and are represented by diamonds. For example, a "Student" entity might have a relationship with a "Course" entity, which could be "enrolls in."

By using an E-R diagram, you can easily see how different pieces of data fit together, ensuring that the database is well-organized and accurately reflects the real-world relationships between data. This makes it easier for database designers and developers to understand the system and create an efficient database structure

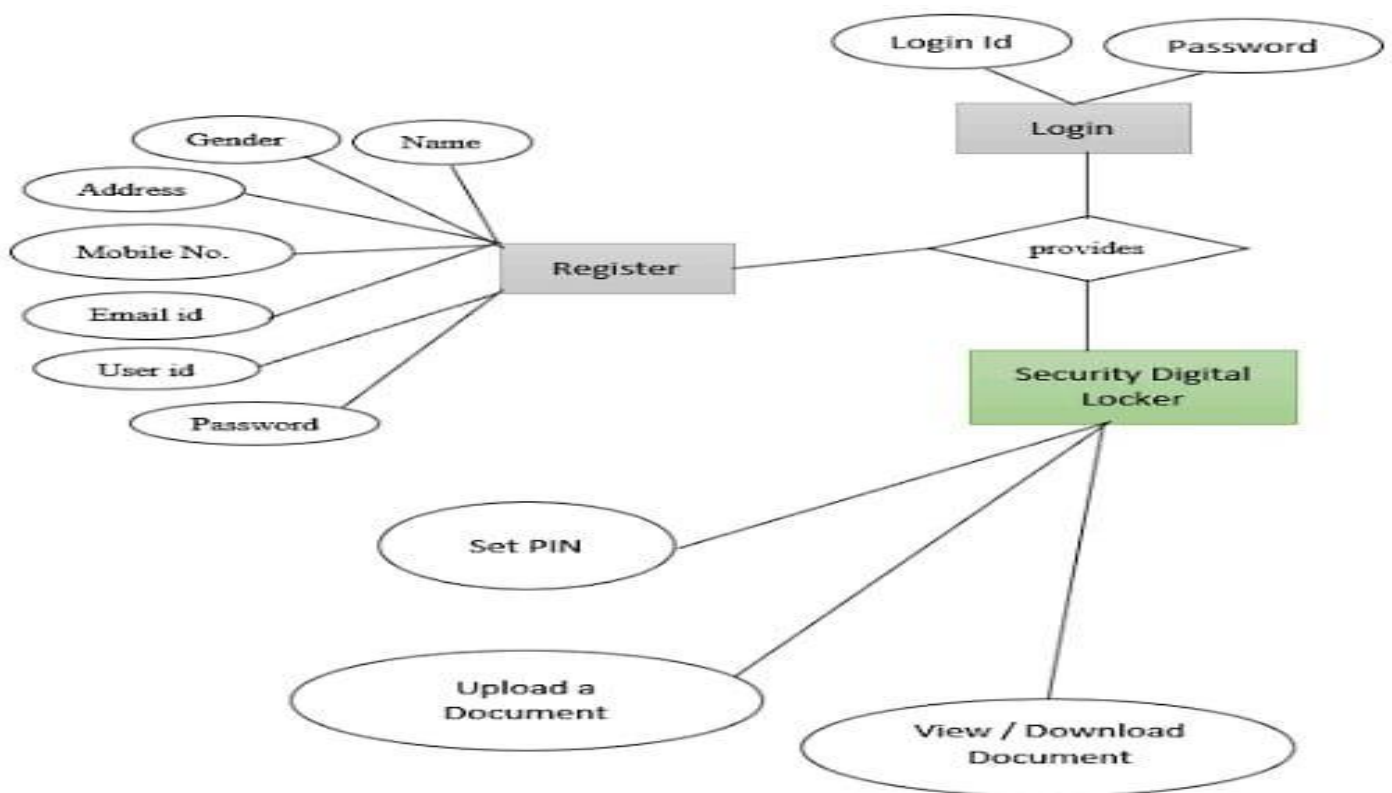


Fig. 2.3 E-R Diagram

In an Entity-Relationship (E-R) diagram, several symbols are commonly used to represent the various components of the diagram. Here are the most used symbols:

Rectangle: Represents an entity. An entity is an object or concept about which data is stored. For example, "Student," "Course," and "Teacher" are entities.

Oval: Represents an attribute. An attribute is a property or characteristic of an entity. For example, "StudentID," "Name," and "DateOfBirth" are attributes of the "Student" entity

Diamond: Represents a relationship. A relationship shows how two entities are related to each other. For example, an "enrolls in" relationship between "Student" and "Course."

Line: Connects attributes to entities and entities to relationships. It indicates the association between different components of the E-R diagram

Double Oval: Represents a multi-valued attribute, which can have multiple values. For example, a "PhoneNumbers" attribute might store more than one phone number for a person.

Dashed Oval: Represents a derived attribute, which can be derived from other attributes. For example, "Age" can be derived from "DateOfBirth."

Double Rectangle: Represents a weak entity, which depends on another entity for its existence. For example, a "Dependent" entity might depend on an "Employee" entity.

Double Diamond: Represents an identifying relationship, which links a weak entity to its owning entity. This is used to show that the weak entity cannot exist without the strong entity

Ellipses: In some variations, attributes can be represented by ellipses instead of ovals.

2.4 SEQUENTIAL DIAGRAM

A sequence diagram is a type of interaction diagram in Unified Modeling Language (UML) that shows how objects interact in a particular sequence to achieve a specific goal or outcome. It is particularly useful for understanding and documenting the dynamic behavior of a system by visualizing the flow of messages, events, and interactions between various components over time. A sequence diagram is a type of chart used to show how different parts of a system interact over time. It helps visualize the order in which events happen and how objects or people (actors) communicate with each other to perform a function.

Here's a simple breakdown:

1. **Lifeline:** A vertical line that represents an object or person involved in the interaction.
2. **Actors:** People or external systems that interact with the system, shown as stick figures.
3. **Messages:** Arrows that show the communication between objects or actors. They indicate who is sending and receiving information.
4. **Activation Bar:** A thin rectangle on the lifeline that shows when an object is doing something.
5. **Fragments:** Boxes that show special conditions like loops or choices (like if-else statements).

Sequence diagrams are useful because they show the step-by-step process of how a task is completed, making it easier to understand and communicate how a system works.

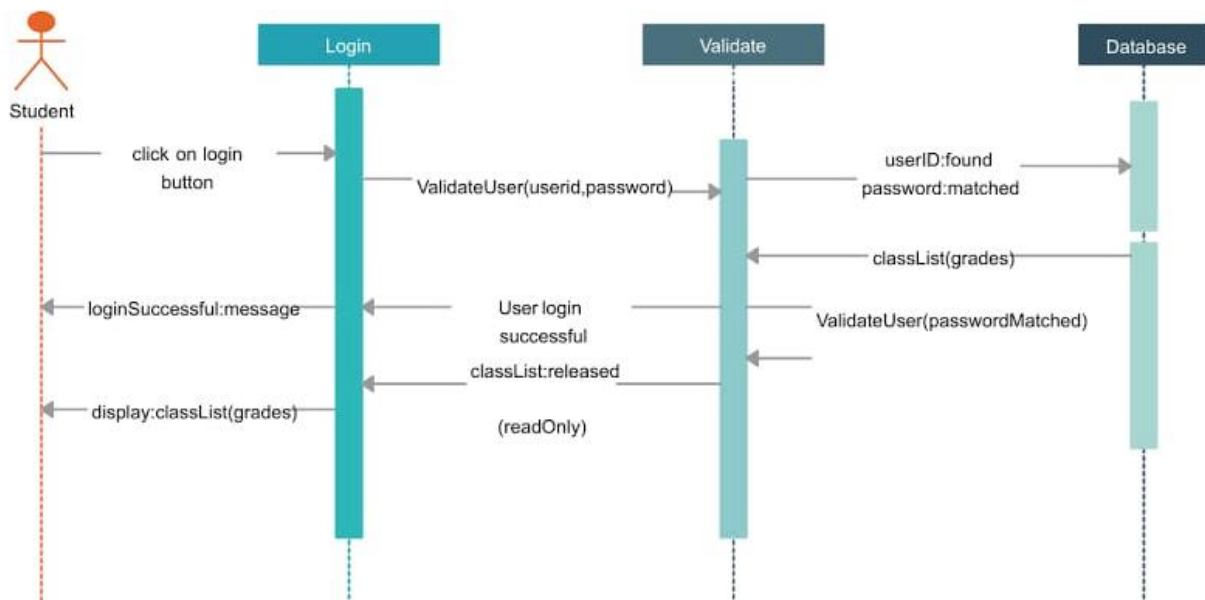


Fig. 2.4 SEQUENTIAL DIAGRAM

3 TECHNOLOGY STACK

3.1 INTRODUCTION

The BVRITH ECAP portal is built using a modern technology stack to ensure robust performance and scalability. The front-end is developed with HTML, CSS, and JavaScript frameworks like React for a responsive and interactive user interface. The back-end is powered by Node.js with Express.js to handle server-side logic, and it uses MongoDB for a flexible, scalable NoSQL database. Additionally, secure online payments are integrated using payment gateway APIs, and the system is hosted on cloud platforms like AWS for reliable access and storage.

3.2 SOFTWARE REQUIREMENTS

Operating System: An operating system (OS) is the software that manages computer hardware and software resources and provides common services for computer programs. It acts as an intermediary between users and the computer hardware, enabling applications to run efficiently and ensuring that different software and hardware components can communicate effectively.

We have used Windows, macOS, or Linux (Ubuntu, CentOS) in our project.

Web Server: A web server is software that hosts websites and delivers web content to users' browsers over the internet. It listens for incoming requests from client browsers, retrieves and processes requested files (like HTML, images, or scripts) from storage, and sends them back to the client. Web servers also manage security protocols, handle sessions, and log access for administrative purposes.

We have used Apache TomCat server 8.5 version in our project.

Database: A database is an organized collection of structured data stored electronically in a computer system. It allows data to be easily accessed, managed, and updated.

Databases use tables to store related information, with each table consisting of rows (records) and columns (fields) that define specific data types and relationships. They are fundamental for storing and retrieving large amounts of information efficiently, supporting various applications from websites to enterprise systems. MySQL is an open-source relational database management system (RDBMS) that uses structured query language (SQL) to manage and manipulate data. It is known for its speed, reliability, and ease of use, making it popular for web applications and small to medium-sized databases. MySQL supports multiple storage engines, transactions, and various data types, offering flexibility for different application needs.

We have used MySQL(structured query language) in our project.

Backend Development: Backend development refers to building and maintaining the server-side logic of web applications. It involves writing code that interacts with databases, manages user authentication, handles requests from the front-end, and ensures the application's functionality and performance.

We have used Java Servlets and JavaServer Pages(JSP) in our project.

JSP stands for JavaServer Pages. It is a technology used for developing dynamic web pages based on Java programming language. JSP allows developers to embed Java code within HTML pages, which is executed on the server side to generate dynamic content that can be sent to the client's web browser.

Frontend Development: Frontend development involves creating and implementing the visible parts of a website or web application that users interact with directly. It focuses on building the user interface (UI) and user experience (UX), using languages like HTML for structure, CSS for styling, and JavaScript for interactive elements. Frontend developers work to ensure the site or app is visually appealing, responsive, and functional across different devices and browsers, while collaborating closely with backend developers to integrate server-side logic and APIs.

HTML, or Hypertext Markup Language, is the standard language used to create and structure content on web pages. It consists of a series of elements or tags that define the different parts of a webpage, such as headings, paragraphs, images, links, and forms. HTML tags are enclosed in angle brackets (< >) and are used to format text, embed multimedia, create lists, and define the overall structure and layout of web documents

Browsers interpret HTML to display content to users, forming the foundation of how information is presented and accessed on the World Wide Web.

CSS, or Cascading Style Sheets, is a language used to define the presentation and appearance of HTML elements on web pages. It controls the layout, colors, fonts, and styling of content, separating the design from the structure and content defined by HTML. CSS allows developers to create consistent and visually appealing web pages, enhancing user experience by customizing the look and feel across different devices and screen sizes. It works by applying rules or styles to HTML elements, using selectors to target specific elements and classes for reusable styles.

We have used HTML5, CSS3, JavaScript in our project.

Development Environment: We have used NetBeans in our project.

NetBeans is an integrated development environment (IDE) primarily used for Java application development, though it supports other programming languages such as PHP, C/C++, and HTML5. It offers features like code editing, debugging, version control, and project management tools within a single user interface. NetBeans is known for its ease of use, extensive plugin ecosystem, and strong community support, making it popular among developers for building and maintaining software projects across different platforms.

3.2 HARDWARE REQUIREMENTS

Processor: A processor, also known as a central processing unit (CPU), is the brain of a computer that executes instructions from software programs. It performs arithmetic, logic, input/output operations, and controls the overall operation of the computer. Processors come in various types and speeds, influencing the computer's performance and capability to handle tasks efficiently.

We have used Multi-core processor (e.g., Intel Core i5/i7 or AMD Ryzen 5/7) in our project.

Memory: Memory in computing refers to electronic components used to store data and instructions temporarily or permanently. It includes primary storage like RAM (Random Access Memory) for fast access to currently running programs and secondary storage like hard drives or SSDs for long-term data storage.

Memory size and speed affect a computer's performance and its ability to handle multiple tasks simultaneously.

We have used Minimum 8 GB RAM, recommended 16 GB or more in our project.

Storage: Storage in computing refers to devices or systems used to hold data permanently or semi-permanently, even when the computer is powered off. It includes hard disk drives (HDDs), solid-state drives (SSDs), and other storage mediums like optical discs and flash drives. Storage capacity and speed vary across different types of storage devices, influencing data access and retrieval times. We have used SSD with at least 256 GB capacity in our project.

Network: A network is a collection of computers and devices interconnected to share resources, information, and services. It allows communication and data exchange between devices using wired or wireless connections. Networks can be local (LAN) within a limited area, wide (WAN) spanning across larger geographical areas, or global (Internet) connecting networks worldwide. We have used Reliable internet connection in our project.

Peripherals: Peripherals are external devices connected to a computer that expand its functionality or provide input/output capabilities. Examples include keyboards, mice, printers, scanners, monitors, and external storage devices like USB drives. Peripherals enable users to interact with and utilize the computer system for various tasks beyond basic processing.

We have used Standard keyboard and mouse, Additional monitors for extended workspace in our project.

3.4 PACKAGS AND MODULES

Packages and modules refer to units of organized code in software development that serve different purposes:

1. **Packages:** In programming languages like Java and Python, packages are collections of related classes and resources bundled together. They help organize code into namespaces, making it easier to manage and reuse components across different parts of a program or between different programs. Packages often encapsulate functionality, providing a way to structure and modularize code for better maintainability and scalability.

2. **Modules:** Modules are similar concepts but are more generalized across various programming languages. They encapsulate a set of related functions, variables, and classes into a single unit. Modules help in organizing code, reducing complexity, and promoting code reusability. In languages like Python, modules are files with Python code that can be imported into other Python scripts to provide additional functionality or resources.

Both packages and modules are fundamental concepts in software development that promote modular programming practices, making codebases more manageable, scalable, and maintainable.

1. Spring Boot (Opinionated framework for building production-ready Spring applications)
2. Hibernate (Object-relational mapping framework)
3. Spring Security (Authentication and access-control framework)
4. JUnit (Testing framework for Java)
5. Apache Tomcat (Web server and servlet container)
6. Bootstrap (Front-end component library)
7. jQuery (JavaScript library for DOM manipulation and AJAX)
8. Popper.js (Dependency for Bootstrap tooltips, popovers, and dropdowns)

4 IMPLEMENTATION

4.1 INTRODUCTION

Implementing BVRITH ECAP involves deploying a comprehensive educational portal designed to streamline student management, academic activities, and administrative processes. Utilizing modern web technologies and a robust backend infrastructure, the implementation focuses on enhancing user experience, ensuring data security, and integrating seamless functionalities like attendance tracking, assignment management, online fee payment, and student profile management. The goal is to create an efficient platform that supports both students and administrators in managing educational operations effectively and fostering a collaborative learning environment

4.2 TEST-CASES

Test cases are detailed instructions or scenarios designed to verify specific functionalities or aspects of a software application. They are written based on requirements and specifications to ensure that the software behaves as expected under various conditions. Each test case typically includes inputs, expected outputs, and steps to execute, along with any preconditions or assumptions necessary for the test. Test cases help testers systematically validate whether the software meets quality standards, identifies defects or discrepancies, and ensures that all aspects of the application are thoroughly tested before deployment.

->If JSP is not retrieving parameters properly, there could be several reasons for this issue:

1. **Incorrect Parameter Names:** Ensure that the parameter names used in the JSP match exactly with the names sent from the previous request (e.g., from a form submission or URL query string).
2. **Encoding Issues:** Check if there are encoding issues causing parameter values not to be interpreted correctly.

Use appropriate character encoding settings (`request.setCharacterEncoding("UTF-8")` for UTF-8 encoding, for example) before retrieving parameters

3. **Encoding Issues:** Check if there are encoding issues causing parameter values not to be interpreted correctly. Use appropriate character encoding settings (`request.setCharacterEncoding("UTF-8")` for UTF-8 encoding, for example) before retrieving parameters.
4. **Scope of Parameters:** Ensure that parameters are being sent and retrieved within the correct scope. Parameters can be passed through request attributes (`request.setAttribute`) or session attributes (`session.setAttribute`) depending on their scope and lifecycle requirements.
5. **Request Forwarding or Redirection:** If parameters are lost during a request forwarding or redirection, ensure that request attributes or session attributes are properly used to pass parameters between different components (like servlets forwarding to JSP).
6. **Servlet Container Configuration:** Check the servlet container (like Tomcat) configuration to ensure that it handles parameter parsing correctly and that there are no server-side configuration issues causing parameter retrieval problems.

->A 404 error typically indicates that the resource or page you are trying to access could not be found on the server. This error is not directly related to database connection issues but rather to the web server not being able to locate the specified URL or resource.

Here are some steps to troubleshoot and resolve a 404 error when creating a database connection in your web application:

1. **Check URL and Configuration:** Ensure that the URL or path specified in your web application for database connection is correct and matches the configuration in your code. Verify the JDBC URL, database name, username, password, and other connection parameters are accurate.
2. **Database Server Availability:** Ensure that the database server (e.g., MySQL, PostgreSQL) is running and accessible from the server where your web application is deployed. Test the database connection separately using database management tools or command line utilities to confirm connectivity.

3. **Servlet Mapping and URL Patterns:** If you are using servlets or JSPs to handle database operations, verify that the servlet mapping or URL patterns in your `web.xml` or annotated configuration (`@WebServlet`) are correctly set up. Incorrect mappings can lead to 404 errors when accessing servlets or JSPs.
4. **Check Server Logs:** Review the server logs (e.g., Tomcat logs, Apache logs) for any specific error messages or exceptions related to the 404 error. Logs can provide insights into what resource or URL is not being found by the server.
5. **File and Directory Permissions:** Ensure that the files and directories required for your web application, including configuration files (`web.xml`, `context.xml`), are deployed correctly and have appropriate read/write permissions.

->If you're encountering an issue where you cannot start or deploy your web application because Tomcat server is already running, here are a few steps to resolve the situation:

1. **Check Running Processes:**

1. First, verify if Tomcat is indeed running. You can check this by opening Task Manager (on Windows) or using commands like `ps` (on Unix-based systems) to list running processes.
2. Look for processes named `java` or specifically `catalina` (Tomcat's executable) to confirm if Tomcat is running.

2. **Stop Tomcat:**

1. If Tomcat is running, you need to stop it before you can start a new instance or deploy your web application.
2. On Windows, you can stop Tomcat by using the Tomcat Monitor application (if installed) or by stopping the service from Services (search for `services.msc` in Start menu).
3. On Unix-based systems, you can stop Tomcat by using the `shutdown.sh` script located in the Tomcat bin directory (`./shutdown.sh`).

3. **Kill the Process:**

1. If you cannot stop Tomcat through normal means (e.g., if it's unresponsive or not stopping), you may need to kill the Java process directly.

4. **Deploy Your Application:**

1. Once Tomcat is restarted and running, deploy your web application by copying the WAR file or the exploded directory structure to Tomcat's webapps directory.

5. **Deploy Your Application:**

1. Once Tomcat is restarted and running, deploy your web application by copying the WAR file or the exploded directory structure to Tomcat's webapps directory.

->The "HTTP Status 404 - Not Found" error typically indicates that the server could not find the requested resource (in this case, the servlet) at the specified URL. Here are some steps to troubleshoot and resolve this issue:

Check Servlet Mapping:

- Ensure that the servlet is correctly mapped in your web.xml deployment descriptor or using annotations (@WebServlet) if you are using Servlet 3.0+ specification.
- **Servlet Class and Package:**
 - Verify that the servlet class (com.example.MyServlet in the example) is correctly defined and located in the specified package.
 - Check for any typos or discrepancies in the servlet class name or package that could prevent the servlet from being found.
- **Context Path:**
 - If you are deploying your application with a context path (e.g., http://localhost:8080/myapp/), make sure that the servlet URL includes the correct context path.
 - For example, if your servlet is mapped to /myServlet, the full URL might be http://localhost:8080/myapp/myServlet.
- **Server Restart:**
 - After making changes to servlet mappings or configurations (web.xml or annotations), restart your servlet container (Tomcat, for example) to ensure that changes are applied correctly.
- **Deployment Issues:**
 - Check the deployment of your application (war file or exploded directory structure) to ensure that the servlet class files are correctly packaged and deployed in the servlet container.

By systematically checking these aspects and ensuring that servlet mappings and configurations are correct, you should be able to resolve the HTTP Status 404 error and access your servlet successfully.

4.3 TESTING

Testing in BVRITH ECAP would involve various aspects to ensure the functionality, reliability, and usability of the educational portal. Here are some key areas where testing would be crucial:

1. **Unit Testing:** Verify individual components like modules, classes, or functions to ensure they perform as expected. This ensures that each part of the system works correctly in isolation.
2. **Integration Testing:** Test how different components work together as a group. This includes testing interactions between modules, database integration, and API interactions within the portal.
3. **System Testing:** Validate the entire system as a whole. This includes testing user flows, navigation, functionality across different browsers and devices, and ensuring that all features and functionalities of BVRITH ECAP work seamlessly together.
4. **User Acceptance Testing (UAT):** Involve end-users (students, faculty, administrators) to test the portal in a real-world environment. This helps identify usability issues, gather feedback, and ensure the portal meets user expectations and requirements.
5. **Performance Testing:** Evaluate how BVRITH ECAP performs under various load conditions. This includes stress testing to determine its stability and responsiveness under peak loads, ensuring it can handle multiple users simultaneously without performance degradation.
6. **Security Testing:** Assess the portal's security measures to identify vulnerabilities and ensure data protection. This includes testing for authentication mechanisms, data encryption, access controls, and protection against common security threats like SQL injection and cross-site scripting (XSS).
7. **Regression Testing:** Ensure that recent code changes or updates do not adversely affect existing functionalities. This involves retesting previously tested features to confirm they still work correctly after modifications.
8. **Accessibility Testing:** Verify that BVRITH ECAP complies with accessibility standards (such as WCAG) to ensure it is usable by individuals with disabilities. Test features like keyboard navigation, screen reader compatibility, and color contrast.

By implementing thorough testing across these areas, BVRITH ECAP can ensure a high level of quality, functionality, and user satisfaction. Testing helps identify and resolve issues early in the development lifecycle, contributing to a reliable and efficient educational portal.

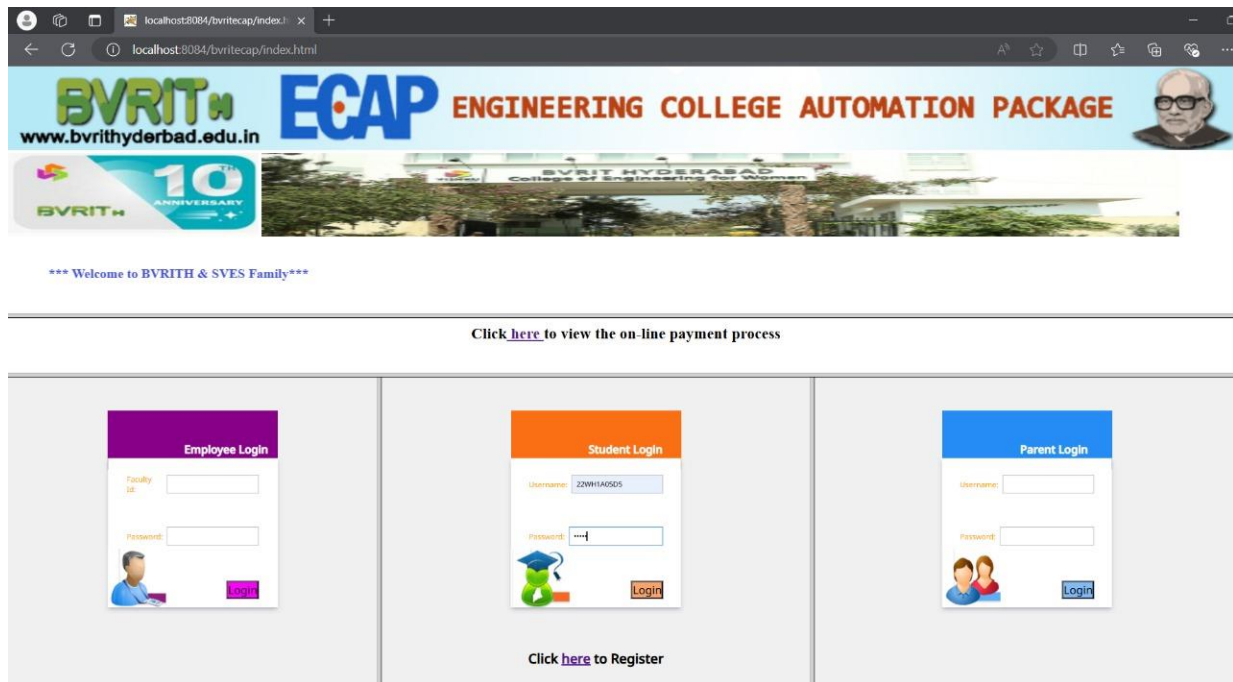


Fig.4.1 login page

The above picture explains as follows:

The BVRITH ECAP login page includes:

- **Header:** BVRITH and ECAP logos, campus image.
- **Welcome Message:** "Welcome to BVRITH & SVES Family".
- **Payment Link:** Online payment process link.
- **Login Sections:** Separate login forms for employees, students, and parents.
- **Registration:** Link to register for new users.

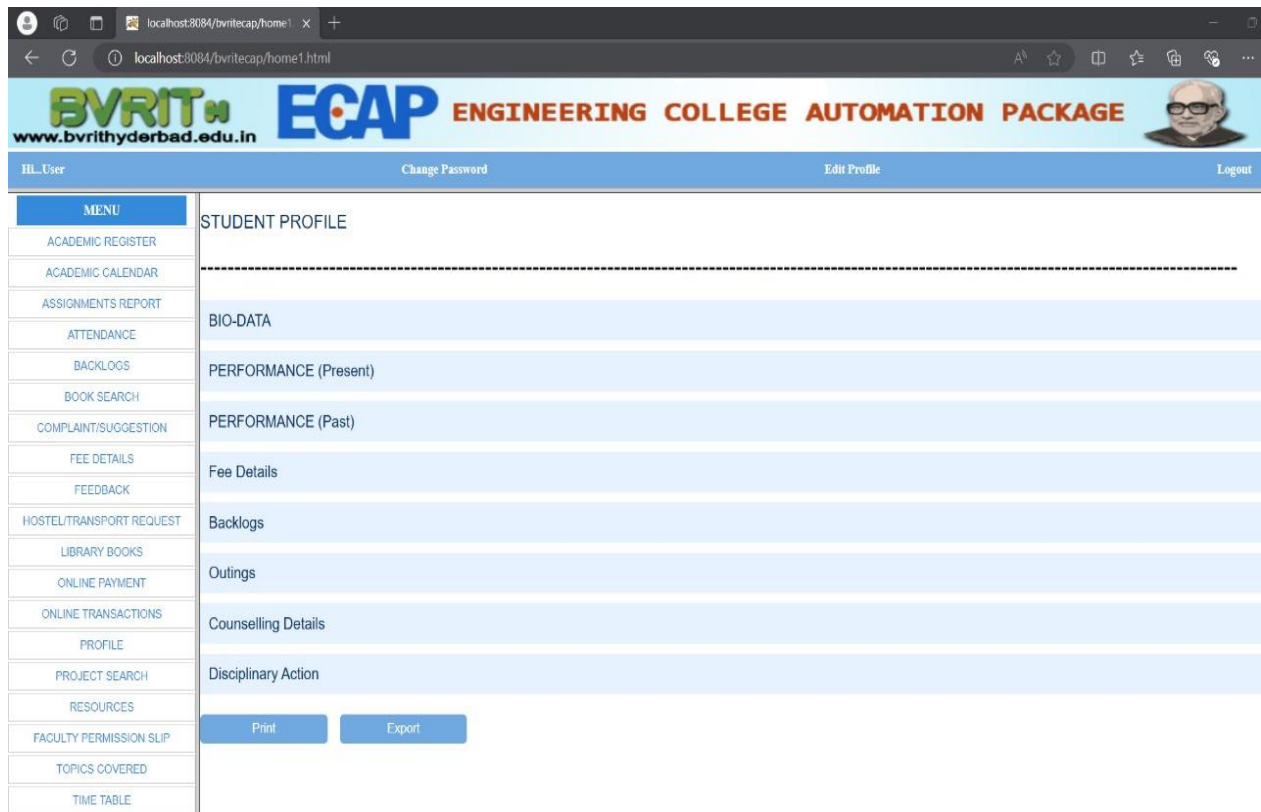


Fig.4.2 Menu Page

In the above page we have created the menu pages by clicking every menu option we can see new thing inside the menu options. We also added new feature called Faculty permission slip. By this we ask permission request to leave. Faculty gets the request. Also we added News Paper. By this we can know the things going to happen.

The screenshot displays the BVRITH ECAP portal interface. At the top, the header includes the BVRITH logo, the ECAP logo, and the text 'ENGINEERING COLLEGE AUTOMATION PACKAGE'. Below the header is a navigation bar with links for 'HL User', 'Change Password', 'Edit Profile', and 'Logout'. A sidebar menu on the left lists various options: MENU, ACADEMIC REGISTER, ACADEMIC CALENDAR, ASSIGNMENTS REPORT, ATTENDANCE, BACKLOGS, BOOK SEARCH, COMPLAINT/SUGGESTION, FEE DETAILS, FEEDBACK, HOSTEL/TRANSPORT REQUEST, LIBRARY BOOKS, ONLINE PAYMENT, ONLINE TRANSACTIONS, PROFILE, PROJECT SEARCH, RESOURCES, FACULTY PERMISSION SLIP, TOPICS COVERED, and TIME TABLE. The main content area is titled 'CHANGE PASSWORD' and contains a form with three input fields: 'Username:', 'Old Password:', and 'New Password:'. Below these fields is a blue button labeled 'Change Password'.

Fig. 4.3 Edit Profile

The screenshot shows the "Change Password" page of the BVRITH ECAP portal.

Elements:

- **Header:** Displays the BVRITH and ECAP logos along with the portal name.
- **Navigation Bar:** Links for "Change Password," "Edit Profile," and "Logout."
- **Sidebar Menu:** Contains various options like Academic Register, Assignments Report, Attendance, Fee Details, and more.
- **Main Content:** "Change Password" form with fields for Username, Old Password, and New Password, and a "Change Password" button.
- **Change Password:** Users can update their passwords by entering their username, old password, and new password, then clicking the "Change Password" button.

BVRIT ECAP ENGINEERING COLLEGE AUTOMATION PACKAGE

www.bvrithyderabad.edu.in

HL User Change Password Edit Profile Logout

MENU

- ACADEMIC REGISTER
- ACADEMIC CALENDAR
- ASSIGNMENTS REPORT
- ATTENDANCE
- BACKLOGS
- BOOK SEARCH
- COMPLAINT/SUGGESTION
- FEE DETAILS
- FEEDBACK
- HOSTEL/TRANSPORT REQUEST
- LIBRARY BOOKS
- ONLINE PAYMENT
- ONLINE TRANSACTIONS
- PROFILE
- PROJECT SEARCH
- RESOURCES
- FACULTY PERMISSION SLIP
- TOPICS COVERED
- TIME TABLE

ON-LINE RECEIPTS

Date Range: 29-05-2024 To: 10-06-2024

FEE DETAILS ☒ College Fee ☐ Exam Fee

Show

SL.NO	ROLL.NO	STUDENT NAME	REF.NO	SBI REF.NO	PAYMENT ON	FEE TYPE	AMOUNT	STATUS	RECEIPT.NO
1	22WH1A05D5	GADDIPATI VEDASRI LAKSHMI	BVRIL96742	IGAPZPVDC8	06/08/2023 20:04:00	Exam Fee(II Sem)	800	Success Check Status	40746
2	22WH1A05D5	GADDIPATI VEDASRI LAKSHMI	BVRIL90426	IGAPCVSLJ3	11/02/2023 10:11:00	Exam Fee	800	Success Check Status	37401
3	22WH1A05D5	GADDIPATI VEDASRI LAKSHMI	BVRIL85049	IGAOURDPC8	14/12/2022 13:28:00	Transport	30000	Success Check Status	34514
4	22WH1A05D5	GADDIPATI VEDASRI LAKSHMI	BVRIL85046		14/12/2022 13:26:00	Transport	30000	Failed	
5	22WH1A05D5	GADDIPATI VEDASRI LAKSHMI	BVRIL84703		13/12/2022 20:25:00	Transport	30000	Failed	

Fig. 4.4 Online Receipt

BVRIT ECAP ENGINEERING COLLEGE AUTOMATION PACKAGE

www.bvrithyderabad.edu.in

HL User Change Password Edit Profile Logout

MENU

- ACADEMIC REGISTER
- ACADEMIC CALENDAR
- ASSIGNMENTS REPORT
- ATTENDANCE
- BACKLOGS
- BOOK SEARCH
- COMPLAINT/SUGGESTION
- FEE DETAILS
- FEEDBACK
- HOSTEL/TRANSPORT REQUEST
- LIBRARY BOOKS
- ONLINE PAYMENT
- ONLINE TRANSACTIONS
- PROFILE
- PROJECT SEARCH
- RESOURCES
- FACULTY PERMISSION SLIP
- TOPICS COVERED
- TIME TABLE

TOPICS COVERED

Operating Systems

Sl.No	Unit.No	Date	No.of Periods	Topics
1	3	28/05/2024	1	semopore
2	1	28/02/2024	6	FUNCTION OS os

Fig. 4.5 Topics Covered

5 CONCLUSION AND FUTURE SCOPE

5.1 CONCLUSION

In conclusion, developing a replica of the BVRITH ECAP website involves meticulous attention to detail in replicating its features such as user authentication, academic data management, and online payment systems. By ensuring fidelity to the original's functionalities and user interface, this replica aims to provide a seamless experience for students, faculty, and administrators alike. Emphasizing robust security measures and scalability, the replica intends to mirror the efficiency and convenience of the original platform, ultimately contributing to an enhanced educational management system for all stakeholders involved.

In summary, the creation of a replica website for BVRITH ECAP necessitates a thorough understanding and replication of its core features, including user-friendly navigation, real-time data updates, and secure transaction capabilities. By faithfully reproducing these elements, the replica aims to maintain continuity in user experience while leveraging modern technologies for improved performance and reliability. This initiative not only facilitates efficient educational management but also underscores the commitment to delivering a seamless digital platform that meets the evolving needs of students, faculty, and administrative staff at BVRITH.

The Conclusion of this project is to design and implement a replica of the BVRITH ECAP website. The Replica should include core functionalities such as student and faculty login, attendance tracking, grade management, timetable scheduling, and communication features. The website should be user-friendly, secure, and responsive

5.2 FUTURE SCOPE

Future enhancements for BVRITH ECAP could include integrating AI-driven analytics for predictive insights into student performance and resource allocation. Expanding mobile compatibility and enhancing user interface design for greater accessibility and usability could also be prioritized.

Additionally, exploring blockchain technology for secure academic credential verification and expanding collaboration tools for remote learning and virtual classrooms could further enrich the platform's capabilities. These advancements aim to keep BVRITH ECAP at the forefront of educational technology, enhancing its effectiveness in supporting both students and faculty in their academic endeavors.

The BVRITH ECAP replica website exemplifies the transformative potential of web-based automation in educational institutions. By optimizing administrative processes and enhancing academic management, the platform promotes operational efficiency and enhances user satisfaction. Future enhancements may include:

- **Advanced Analytics:** Integration of data analytics tools to derive actionable insights for informed decision-making and strategic planning.
- **Mobile Application Development:** Developing mobile-responsive versions or dedicated apps to extend platform accessibility and usability.
- **AI-driven Features:** Incorporation of artificial intelligence algorithms for predictive analytics, personalized learning experiences, and automated decision support systems.

These advancements will further propel the BVRITH ECAP replica website towards achieving its mission of fostering a technologically empowered educational environment.

6 REFERENCES

- Oracle. (n.d.). *JDBC Developer's Guide*. Retrieved from <https://docs.oracle.com/en/database/oracle/oracle-database/19/jjdbc/index.html>
- Oracle. (n.d.). *Java SE Documentation*. Retrieved from <https://docs.oracle.com/javase/8/docs/>
- Oracle. (n.d.). *Java EE Documentation*. Retrieved from <https://javaee.github.io/tutorial/>
- W3C. (n.d.). *HTML & CSS Standards*. Retrieved from <https://www.w3.org/standards/webdesign/htmlcss>
- Oracle. (n.d.). *MySQL Documentation*. Retrieved from <https://dev.mysql.com/doc/>
- Duckett, J. (2011). *HTML and CSS: Design and Build Websites*. John Wiley & Sons.
- Nixon, R. (2012). *Learning PHP, MySQL, JavaScript, and CSS: A Step-by-Step Guide to Creating Dynamic Websites*. O'Reilly Media.