







**A**

**PROJECT REPORT**

**On**

**“EdifyLabs Business Tool”**

**Submitted By**

**Tanmay Kisan Barvi (5053)**

**Under the guidance of  
Prof. Milind Deshkar**

**SUBMITTED TO**



**Savitribai Phule Pune University**

**As a partial fulfilment for the award of the degree of**

**MASTER OF COMPUTER APPLICATIONS**

**At**

**ASM'S Institute of Business Management and Research,  
MCA Institute, Chinchwad, Pune- 411019  
A.Y 2022-2023**





### **ACKNOWLEDGEMENT**

It is a pleasure to acknowledge the assistance of several people and institutions in this effort. This project has turned me into a debtor. First and foremost, I feel indebted to my guide **Prof. Milind Deshkar** and all the faculty members **of Department of MCA, ASM's IBMR, MCA Institute Chinchwad** for their valuable guidance, continuous support and advice and constant encouragement throughout my project work.

I would like to extend my gratitude to **honorable Dr. V. P. Pawar Sir, Director, ASM's IBMR, MCA Institute, Chinchwad** for being a constant source of inspiration.

Finally, I would like to extend my thanks to all those who have contributed, directly or indirectly, to make this project successful.

Tanmay Kisan Barvi (5053)

<b>Chapter No</b>	<b>Details</b>	<b>Page No.</b>
<b>1</b>	<b>Introduction</b>	1
1.1	Company Profile	2
1.2	Abstract	4
1.3	Existing System and Need for System	5
1.4	Scope of System	7
1.5	Operating Environment - Hardware and Software	8
1.6	Brief Description of Technology Used	9
<b>2</b>	<b>Proposed System</b>	13
2.1	Study of Similar Systems	14
2.2	Feasibility Study	16
2.3	Objectives of Proposed System	17
2.4	User Requirements	19
<b>3</b>	<b>Analysis and Design</b>	21
3.1	System Requirements (Functional and Non-Functional requirements)	22
3.2	Data Flow Diagram (DFD)	28
3.3	Entity Relationship Diagram (ERD)	34
3.4	Table Structure	35
3.5	Use Case Diagrams	40
3.6	Activity Diagram	41
3.7	Sequence Diagram	42
3.8	Deployment Diagram	43
3.9	Component Diagram	44
3.10	Sample Input and Output Screens	45
3.11	Report	49
<b>4</b>	<b>Coding</b>	51
4.1	Algorithms	52

---

4.2	Code snippets	54
<b>5</b>	<b>Testing</b>	63
5.1	Test Strategy	64
5.2	Test Case / Test Script	66
5.3	Defect report / Test Log	72
<b>6</b>	<b>Limitations of Proposed System</b>	75
<b>7</b>	<b>Proposed Enhancements</b>	77
<b>8</b>	<b>Conclusion</b>	79
<b>9</b>	<b>Bibliography</b>	81
<b>10</b>	<b>User manual</b>	83



## **CHAPTER 1**

# **Introduction**

## 1.1 Client's Profile

### **EdifyLabs**

**COMPANY NAME :** Edify Labs



#### **ABOUT COMPANY :**

Edify Labs in Thane, Mumbai is one of the leading businesses in the Career Counseling Canfers. Also known for Career Counseling Canfers, Aptitude Test Canfers, Life Coaching Services, Corporate Training Services, Motivational Activity Organizers, Personal Coaching and much more.

#### **FOUNDER:**

Founded by Nitesh Gawade, is an Multiple Award Winning Strategic Learning Partner to India's Top Corporates and Best Selling Author. Nitesh has trained more than 2,90,000 participants across India in the last 10 years. Nitesh is a coach to authors , entrepreneurs and potential CXOs.

## **SERVICES OFFERED (COURSES) :**

Edify Labs offers personalized coaching programs such as "Make It Happen - Personal Coaching," certification courses like "Licensed Purpose Coach," and a coaching program for aspiring teenage entrepreneurs called the "Teenpreneur Fellowship Personal Coaching Program." They also provide courses in career guidance, soft skills training, leadership development, personality development, interview preparation, team building, and motivational workshops. The courses offered by Edify Labs are priced in the range of 1 lakh to 7 lakhs, depending on the program and its duration.

## **DETAILS:**

Website : [www.livingtheunlived.com](http://www.livingtheunlived.com)

Email : [nitesh.gawade@edifylabs.com](mailto:nitesh.gawade@edifylabs.com) Customer Care : 80 80 024 024

Address : EDIFY LABS , Fourth Floor, WEWORK Zenia, Hiranandani Estate, Thane West.

## 1.2 Abstract

The Edify Labs HR Software System is a comprehensive and user-friendly solution designed to streamline HR processes and enhance efficiency in growing organizations. With advancements in technology and the increasing complexity of HR tasks, there is a need for a personalized and data-driven tool that can simplify and automate HR activities. Edify Labs aims to fill this gap by providing HR professionals with an easy-to-use software system that offers a tailored and comprehensive HR experience.

The software system is built on the Python Django web framework, leveraging its robust features and flexibility. It offers various modules and functionalities to meet the diverse needs of HR professionals and their clients. The system includes an HR dashboard for managing client information and performing CRUD operations. Scheduling capabilities enable users to schedule meetings and appointments, ensuring effective time management. The filtration feature allows for easy searching and management of data across different sections.

Data visualization tools provide graphical representations of HR-related information, enabling users to track and analyze data effectively. The report generation feature enables HR professionals to generate customized reports for record-keeping and analysis. Invoicing functionality facilitates the creation of payment invoices for clients, supporting various payment methods.

The system ensures secure authentication and data protection measures, allowing authorized users to access the system and ensuring the confidentiality of sensitive HR information. Integration with databases and external networks ensures accurate and up-to-date information availability for HR professionals.

## 1.3 Existing System and Need for System

### **Existing System:**

The existing HR software systems often lack personalization and tailored experiences for individual clients. This means that HR professionals may struggle to effectively address the specific needs and requirements of each client they work with. The lack of customization can limit their ability to provide a high level of service and support to clients.

Furthermore, some HR software systems can be complex and difficult to navigate. This complexity may result in a steep learning curve for HR professionals, leading to reduced efficiency and productivity. The user interface may not be intuitive, making it challenging for HR professionals to quickly perform tasks and access the information they need.

Another limitation of many existing HR software systems is the lack of in-depth data analysis capabilities. While these systems may store and manage data related to clients' HR activities, they may not provide robust analytical tools to gain insights from this data. Without access to comprehensive data analysis features, HR professionals may struggle to identify trends, patterns, and opportunities for improvement within their clients' HR processes.

To overcome these limitations, there is a need for a more personalized, user-friendly, and data-driven HR software solution. A modern HR software system should allow for customization and flexibility, enabling HR professionals to tailor the system to the unique needs of each client. This could involve customizing workflows, reports, and user interfaces to align with specific client requirements.

## **Need for System:**

Existing HR software systems have proven to be valuable tools for managing clients data and related activities. However, they often fall short in meeting the evolving needs of HR professionals and fail to provide a comprehensive solution for optimizing HR processes. This creates a compelling need for a new and innovative HR software system, such as Edify Labs, to address the following key challenges:

**Lack of Personalization:** Existing systems often lack the ability to provide personalized experiences for individual clients. HR professionals require a system that can tailor its functionalities and features to meet the unique needs of clients, enabling them to effectively address their concerns and provide a more engaging HR experience.

**Complex and Inefficient User Interfaces:** Some HR software systems suffer from complex user interfaces and navigation structures, leading to reduced efficiency and productivity. HR professionals need a user-friendly system with intuitive interfaces that simplify tasks, streamline processes, and enhance overall usability.

**Limited Data Analysis Capabilities:** Existing systems may lack advanced data analysis features, preventing HR professionals from gaining valuable insights into employee behavior, performance, and engagement. A new HR software system should provide robust data analysis tools and visualizations to empower HR professionals with actionable intelligence for informed decision-making.

**Ineffective Process Improvement:** Without comprehensive data analysis and reporting capabilities, HR professionals may struggle to identify areas for process improvement and efficiency gains. A new system should enable HR professionals to identify trends, patterns, and bottlenecks within HR processes, facilitating continuous improvement initiatives.

## 1.4 Scope of System

The scope of the Edify Labs Business Tool project encompasses the development of a web-based application that specifically caters to the needs of HR departments. The primary objective is to provide HR professionals with an intuitive and user-friendly solution for efficiently managing client details, course scheduling, and various related activities. The application will feature an HR dashboard that serves as a centralized hub for accessing and organizing client information. HR professionals will be able to add, update, and manage client details effortlessly, ensuring accurate and up-to-date records. The system will also include advanced filtration capabilities, enabling HR professionals to search and manage client data with ease. Data visualization will play a crucial role in the application, allowing HR professionals to gain insights from the collected data. Graphs, charts, and other visual representations will provide a comprehensive overview of key HR metrics, empowering professionals to make informed decisions and identify trends or patterns.

The reporting functionality of the system will enable HR professionals to generate customized reports, ensuring that crucial information is readily available for analysis and record-keeping purposes. This feature will streamline the process of generating reports, saving time and effort for HR professionals.

The application will also support payment invoicing, allowing HR professionals to generate invoices for clients who have made payments through various methods such as RTGS, UPI, or NFT. Additionally, the system will facilitate data downloading, enabling HR professionals to extract required data in formats such as CSV for further analysis or integration with external tools.

## **1.5 Operating Environment - Hardware and Software**

### **HARDWARE REQUIREMENTS**

- Operating System: operates on multiple platforms, including Windows, macOS, and Linux
- Web Browser: Google Chrome, Mozilla Firefox, or Safari.
- RAM: 4GB +
- Storage: 1-2 GB

### **SOFTWARE REQUIREMENTS**

- FRONTEND: HTML5 , CSS3 , JavaScript , Bootstrap
- BACKEND: Python, Django web framework, ORM
- DATABASE: SQLite, Postgres
- EDITOR: VScode



## 1.6 Brief Description of Technology Used

### Operating System:

- The HR software application is designed to be compatible with multiple operating systems such as Windows, macOS, and Linux. This compatibility ensures that HR professionals can use the application regardless of their preferred platform, increasing accessibility and usability.
- The application leverages platform-specific features and APIs to provide a seamless user experience and take advantage of system capabilities.

### Web Browser:

- The application is developed to work seamlessly on popular web browsers such as Google Chrome, Mozilla Firefox, and Safari. Compatibility with these browsers ensures that HR professionals can access and use the application without any compatibility issues, enhancing user experience and convenience.
- The application utilizes modern web standards, adheres to W3C recommendations, and incorporates progressive enhancement techniques to ensure broad browser compatibility.

### RAM:

- The recommended minimum RAM of 4GB is based on the application's resource requirements and performance considerations. Sufficient RAM ensures smooth and efficient operation, allowing the application to handle complex operations, process large data sets, and respond quickly to user interactions.
- With ample RAM, the application can cache frequently accessed data, optimize memory usage, and deliver a responsive user experience.

### Storage:

- The allocated 1-2 GB of storage space is designed to accommodate the storage needs of the application. It allows for the storage of necessary files, databases, and related assets, ensuring seamless operation and preventing issues related to insufficient disk space.
- The application may store user data, documents, media files, configuration files, and other resources, providing a comprehensive and scalable solution for HR professionals' needs.

### Frontend:

The frontend of the application utilizes HTML5, CSS3, JavaScript, and Bootstrap to create a modern and interactive user interface.

- HTML5 provides a rich set of elements, APIs, and semantic markup, allowing for the structuring and presentation of content effectively.
- CSS3 offers extensive styling capabilities, including advanced layout techniques, animations, and responsive design features, resulting in a visually appealing and consistent UI.
- JavaScript adds interactivity and dynamic functionality to the frontend, enabling features such as form validation, data manipulation, real-time updates, and client-side data processing.
- Bootstrap, as a CSS framework, provides pre-designed components, grids, and responsive utilities, ensuring a consistent and mobile-friendly UI across different devices and screen sizes.

#### Backend:

- Python serves as the backend programming language due to its simplicity, readability, and extensive ecosystem of libraries and frameworks.
- Django, a powerful Python web framework, is utilized to build the backend infrastructure of the application. It offers a high-level abstraction for handling common web development tasks, including URL routing, request handling, database management, and user authentication.
- Django's ORM simplifies database interactions by mapping database tables to Python objects, reducing the need for manual SQL queries and enhancing productivity.
- The combination of Python and Django provides a robust, secure, and scalable backend foundation for the HR software application.

#### Database:

- The HR software application supports both SQLite and Postgres as database management systems.
- SQLite is a lightweight, file-based database suitable for small-scale applications. It is self-contained, requiring no separate server setup, and simplifies deployment and management.
- Postgres, on the other hand, is a feature-rich and scalable relational database management system suitable for larger applications. It offers advanced database features, performance optimizations, and support for concurrent access and high data volumes.
- The choice of database systems provides flexibility based on the application's scale, performance requirements, and specific use cases.

**Editor:**

- Visual Studio Code (VScode) is recommended as the code editor for developing and customizing the HR software application.
- VScode offers a rich set of features, including syntax highlighting, code completion, linting, debugging, and version control integration.
- The extensibility of VScode allows HR professionals to customize their development environment with various extensions, enhancing productivity and collaboration.
- With support for multiple programming languages, including Python, VScode provides an efficient and user-friendly code editing experience.

**Jasmine:**

- Jasmine is a popular JavaScript testing framework that can be integrated into Django applications to enable automated testing of frontend JavaScript code.
- With Jasmine, developers can write test cases and specifications to verify the behavior and functionality of JavaScript code used in the frontend of the Django application.
- Jasmine provides a syntax that resembles natural language, making it readable and expressive.
- By using Jasmine in Django, developers can ensure the correctness and robustness of the frontend JavaScript code, which contributes to the overall stability and quality of the HR software application.

## **CHAPTER 2**

# **Proposed System**

## 2.1 Study of Similar Systems

Sr No.	Refereed From	Application Name	Name of Modules	How it Works
1	HRMS Guide	Workday	Core HR, Payroll, Benefits, Time and Attendance, Talent Management	Workday offers a comprehensive suite of modules for managing core HR processes, payroll, benefits administration, time and attendance tracking, and talent management. The Core HR module handles employee data, including personal information, employment history, and organizational structure. The Payroll module automates payroll processing, tax calculations, and reporting. Benefits administration facilitates the management of employee benefits and open enrollment processes. Time and Attendance module tracks employee attendance, leave accruals, and absence management. Workday's cloud-based solution integrates these modules, providing HR professionals with a unified platform for managing HR processes.
2	G2	BambooHR	Employee Database, Time Off Tracking, Performance Management, Recruitment, Onboarding	BambooHR focuses on core HR functions and employee management. The Employee Database module allows HR professionals to maintain employee records, including personal information, employment history, and documents. Time Off Tracking simplifies leave management by providing a centralized

				<p>system for requesting, approving, and tracking time off. The Performance Management module helps streamline performance evaluations, goal setting, and feedback processes. The Recruitment module assists in job posting, applicant tracking, and interview scheduling. The Onboarding module ensures a smooth transition for new hires by automating the onboarding process and managing necessary documentation.</p>
3	Capterra	ADP Workforce Now	Payroll, Time and Attendance, Benefits Administration, HR Management	<p>ADP Workforce Now is a comprehensive HR software solution designed for small to mid-sized businesses. The Payroll module handles payroll processing, tax calculations, and direct deposit management. Time and Attendance module allows employees to clock in/out, tracks attendance, and provides scheduling capabilities. The HR Management module covers employee records management, onboarding, performance management, and compliance tracking. ADP Workforce Now offers integration with external systems and provides self-service features for employees and managers. The system streamlines HR processes and ensures compliance with payroll and HR regulations.</p>

## 2.2 Feasibility Study

There are mainly three kind of feasibility study that are equally important for this report:

- Technical Feasibility
- Economic Feasibility
- Operational Feasibility

### Operational Feasibility

The operational feasibility of Edify Labs Business Tool is high. The system will provide a userfriendly and intuitive interface for HR professionals to manage their clients' HR-related data and activities. The system will be designed to automate and streamline HR-related tasks, such as scheduling, data visualization, and report generation, saving time and improving efficiency. The system will also be designed to be customizable, allowing users to tailor the system to their specific needs and preferences.

### Economic Feasibility:

The economic feasibility of Edify Labs Business Tool is high. The system will provide a costeffective HR solution for growing organizations. The system will be designed to reduce manual HR-related tasks, such as data entry and report generation, which can save time and improve productivity. The system will also be designed to reduce the need for manual labor, such as HR administrative staff, which can result in cost savings for organizations. Additionally, the system will be priced competitively, making it accessible for small and medium-sized businesses.

### Technical Feasibility

The technical feasibility of Edify Labs Business Tool is high. The system will be built on the Python Django web framework, which is a reliable and widely-used framework for building web applications. The system will be designed to be scalable, modular, and flexible, allowing for future updates and modifications. The system will also be designed to integrate with other HR-related software, such as payroll management software, to provide a comprehensive HR solution.



## 2.3 Objectives of Proposed System

The objective of Edify Labs Business Tool is to provide HR professionals with an easy-to-use web-based platform to manage their clients' information, courses, scheduling, and related activities. The platform is designed to streamline HR processes, reduce administrative workload, and increase efficiency, making it easier for HR professionals to get their work done in less time. Additionally, the platform is intended to provide valuable insights into client data, helping Edify Labs to make informed decisions and drive growth.

The objective of the Edify Labs Business Tool is to empower HR professionals by providing them with a comprehensive and user-friendly web-based platform to manage their clients' information and related activities. The tool aims to simplify HR processes and streamline operations, ultimately reducing administrative workload and increasing overall efficiency.

By offering features such as an HR dashboard, scheduling, filtration, data visualization, report generation, payment invoicing, and data downloading, the platform equips HR professionals with the necessary tools to effectively manage their clients' information. The HR dashboard serves as a centralized hub where HR professionals can access and update client details, track progress, and perform various CRUD (Create, Read, Update, Delete) operations.

The scheduling feature enables HR professionals to efficiently manage course schedules and appointments with clients. They can easily view, create, and modify schedules, ensuring that client sessions and meetings are organized and well-coordinated. This helps in optimizing time management and ensuring effective communication with clients.

The platform's data visualization capabilities provide HR professionals with valuable insights into client data. By presenting data in visually appealing graphs and charts, the platform facilitates the analysis of trends, patterns, and performance metrics. These insights enable HR professionals to make data-driven decisions, identify areas for improvement, and tailor their services to meet client

needs more effectively.

Report generation functionality allows HR professionals to generate comprehensive reports on client activities, progress, and performance. These reports serve as a documentation tool, providing a clear overview of client engagements and outcomes. HR professionals can leverage these reports for internal purposes, client communication, and performance evaluation.

Payment invoicing features enable HR professionals to generate invoices for clients who have made payments for services rendered. This simplifies the billing process and ensures accurate financial records. Additionally, the platform supports downloading data in various formats, such as CSV, facilitating further analysis and integration with external tools.

## 2.4 User Requirements

- **User-Friendly Interface:** The software system should have an intuitive and user-friendly interface that is easy to navigate and understand, even for users with limited technical expertise.
- **Client Management:** The system should provide a centralized hub, such as an HR dashboard, for HR professionals to efficiently manage client information. It should allow users to add, update, and organize client details effortlessly.
- **CRUD Operations:** The system should support CRUD (Create, Read, Update, Delete) operations for managing client information, enabling HR professionals to easily perform necessary actions on client data.
- **Scheduling Capabilities:** The system should have scheduling features that allow HR professionals to schedule meetings, appointments, and other HR-related activities. This functionality should support effective time management and enable users to set reminders.
- **Data Filtration:** The system should include advanced filtration capabilities that enable HR professionals to search and manage client data efficiently. Users should be able to filter data based on specific criteria and easily retrieve the desired information.
- **Data Visualization:** The system should provide data visualization tools, such as graphs, charts, and other visual representations, to present HR-related information in a graphical format. This functionality should help HR professionals track and analyze data effectively and identify trends or patterns.

- **Customized Reporting:** The system should allow HR professionals to generate customized reports for analysis and record-keeping purposes. Users should be able to select specific data fields and generate reports in a format that is easy to understand and share.
- **Invoicing Functionality:** The system should support the creation of payment invoices for clients. It should facilitate the generation of invoices using various payment methods, such as RTGS, UPI, or NFT, and ensure accuracy in billing and payment tracking.
- **Data Security:** The system should implement secure authentication and data protection measures to ensure that only authorized users can access the system. It should also ensure the confidentiality of sensitive HR information.
- **Integration and Data Availability:** The system should integrate with databases and external networks to ensure accurate and up-to-date information availability for HR professionals. It should support data synchronization and provide seamless integration with external tools, if required.
- **Data Downloading:** The system should allow HR professionals to download required data in formats such as CSV for further analysis or integration with external tools. This functionality should enable users to extract and manipulate data as needed.

# **CHAPTER 3**

# **Analysis and Design**

### 3.1 System Requirements (Functional and Non-Functional requirements)

#### Functional Requirements:

Use Case Name	Id	Trigger	Precondition	Basic Path	Post Condition
User-Friendly Interface	UC-001	HR professional wants to navigate the system easily.	HR professional has appropriate system access	1. HR professional logs into the system . 2. HR professional is presented with an intuitive and user-friendly interface. 3. HR professional can easily navigate through different sections and modules of the system	HR professional can perform tasks and access information efficiently.
Data Analysis	UC-002	HR professional wants to gain insights from HR-related data.	HR professional has appropriate system access and relevant data available.	1. HR professional logs into the system. 2. HR professional selects the desired data for analysis, such as employee	HR professional gains valuable insights from the HR data.

				performance, leave records, or training data. 3. HR professional uses the data analysis tools provided by the system to derive insights, identify trends, and uncover areas of improvement.	
Data Filtration	UC-003	HR professional wants to filter and search HRrelated data.	R professional has appropriate system access and data available for filtration.	1. HR professional selects the desired data section or module. 2. HR professional specifies the filtration criteria, such as client name, date range, or payment status. 3. The system filters the data HR professional can efficiently search and manage HR data based on	HR professional can efficiently search and manage HR data based on specific criteria.

				specific criteria on the specified criteria and presents the filtered results to the HR professional	
Reporting	UC-004	HR professional wants to generate customized reports.	HR professional has appropriate system access and relevant data available.	1. HR professional selects the desired data or report template. 2. HR professional customizes the report parameters, such as data fields, date range, or clientspecific information. 3. The system generates the customized report, incorporating the specified parameters and data.	HR professional obtains a customized report that can be used for analysis, recordkeeping, or sharing with clients.
Scheduling Meetings	UC-005	HR professional needs to	HR professional as appropriate	HR professional navigates to the	The meeting appointment is



		schedule meetings or appointments	system access and meeting details	scheduling module. 2. HR professional enters the meeting details, including date, time, participants, and location. 3. The system checks for conflicts and availability of participants. 4. If there are no conflicts, the system saves the meeting details and sends notifications to the participants.	successfully scheduled, and participants are notified about the details.
Customization	UC-006	HR professional wants to customize the system for a specific client	HR professional has appropriate system access and client information.	1. HR professional logs into the system. 2. HR professional selects the client they want to customize the system for. 3. HR professional identifies the	The system is customized according to the specific needs of the client

				customization requirements based on client needs. 4. HR professional modifies workflows, reports, and user interfaces to align with client requirements	
--	--	--	--	---	--

## Non-Functional Requirements:

### Usability:

The system should have a user-friendly interface with intuitive navigation and clear instructions. It should be easy for HR users to learn and use the system efficiently.

### Performance:

The system should provide fast response times, ensuring a smooth user experience even with a large number of clients and data.

It should handle concurrent user requests and database transactions efficiently.

### Security:

The system should implement robust security measures to protect client data and user authentication. User access should be restricted based on roles and permissions to maintain data confidentiality.

**Scalability:**

The system should be designed to handle an increasing number of clients, courses, and user accounts. It should be scalable to accommodate future growth and additional features.

**Reliability:**

The system should be highly reliable, with minimal downtime and data loss.

Regular backups and data recovery mechanisms should be in place to ensure data integrity.

**Compatibility:**

The system should be compatible with common web browsers such as Google Chrome, Mozilla Firefox, and Safari. It should be responsive and accessible on different devices, including desktops, laptops, and tablets.

**Data Privacy:**

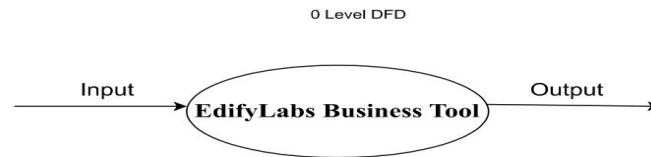
The system should comply with data privacy regulations and ensure the secure storage and handling of client information.

Personal data should be encrypted and protected against unauthorized access.

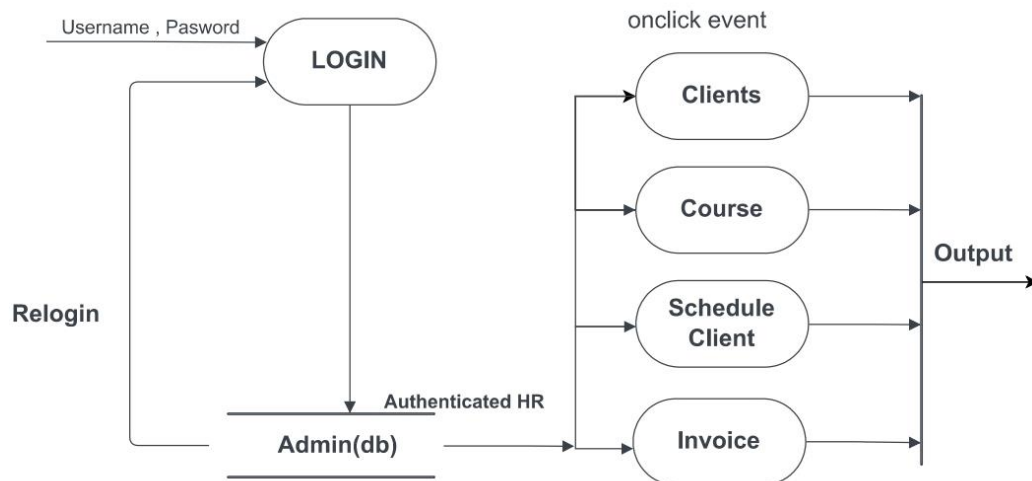
**Documentation and Support:**

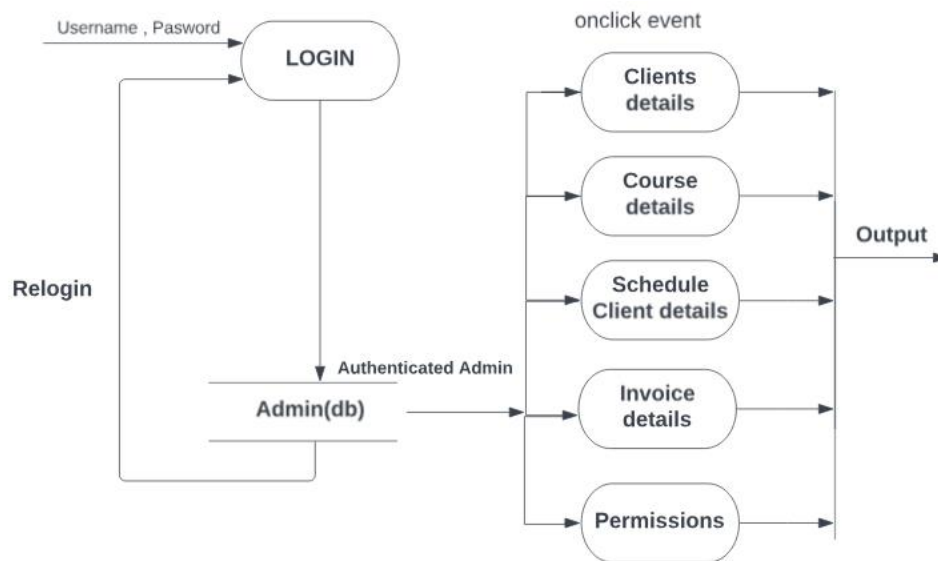
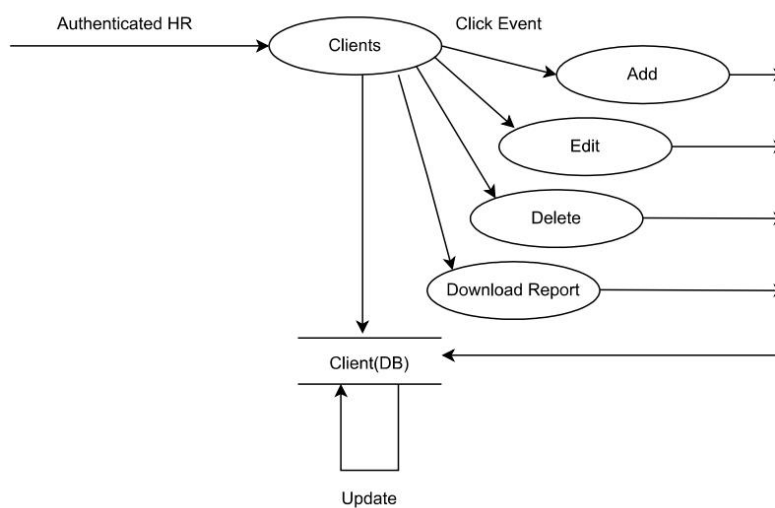
The system should have comprehensive documentation to guide users on its usage, features, and troubleshooting. Adequate support channels, such as email or online chat, should be available to assist users in case of any issues or inquiries.

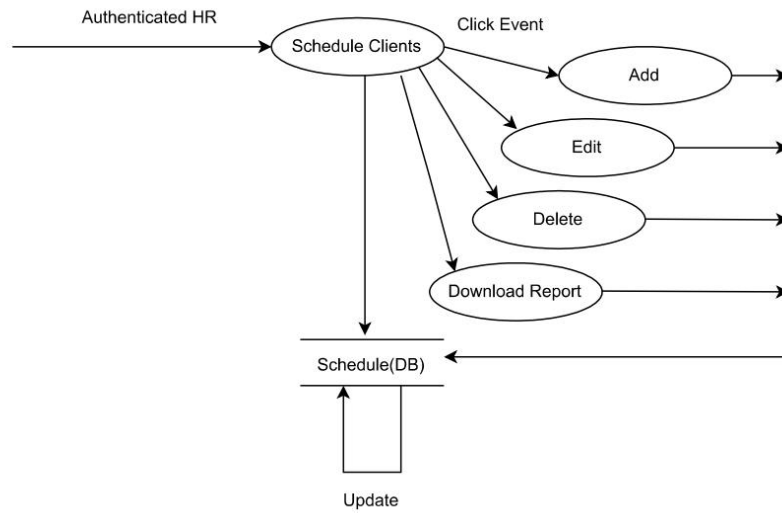
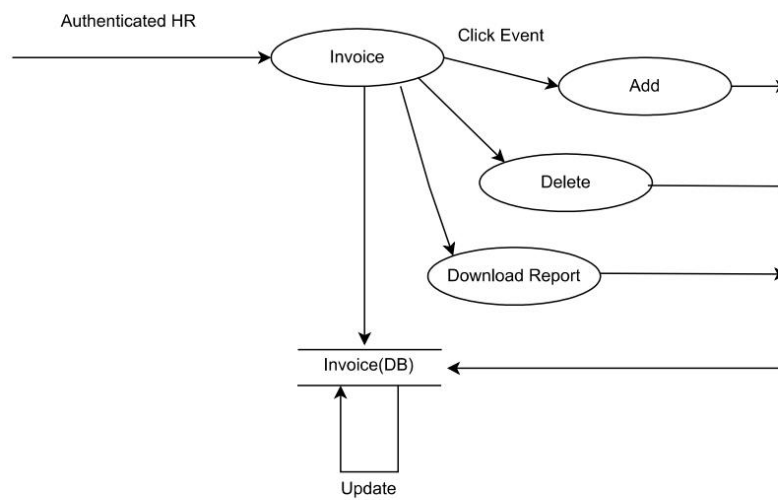
## 3.2 Data Flow Diagrams (DFD)

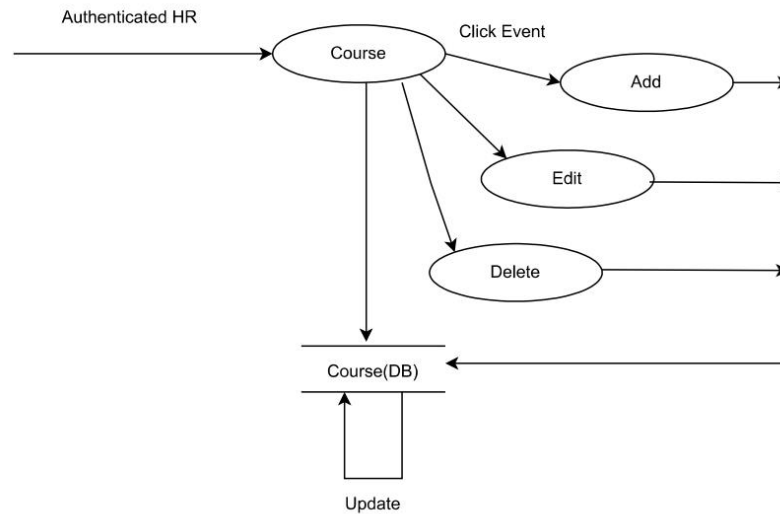
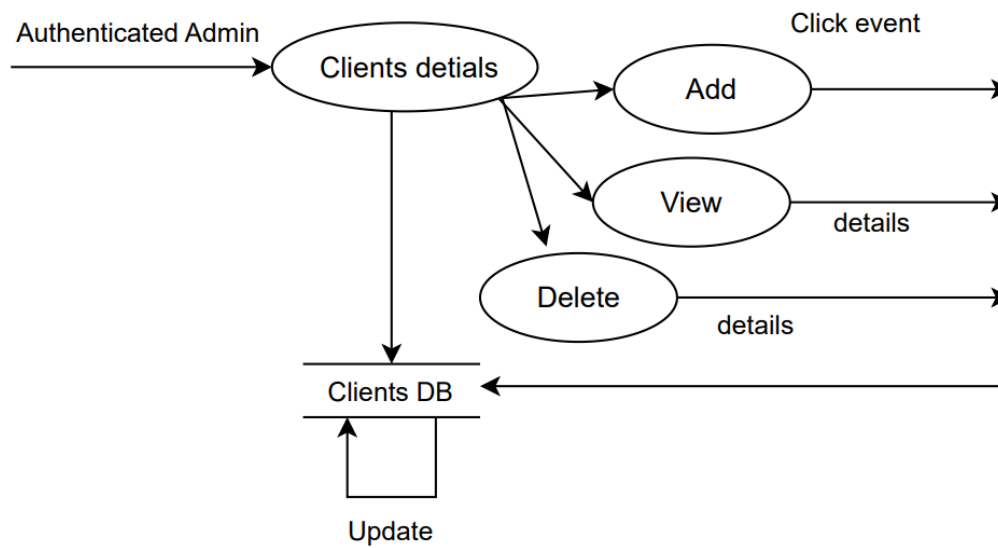


### First level Dataflow diagram (1 (a))

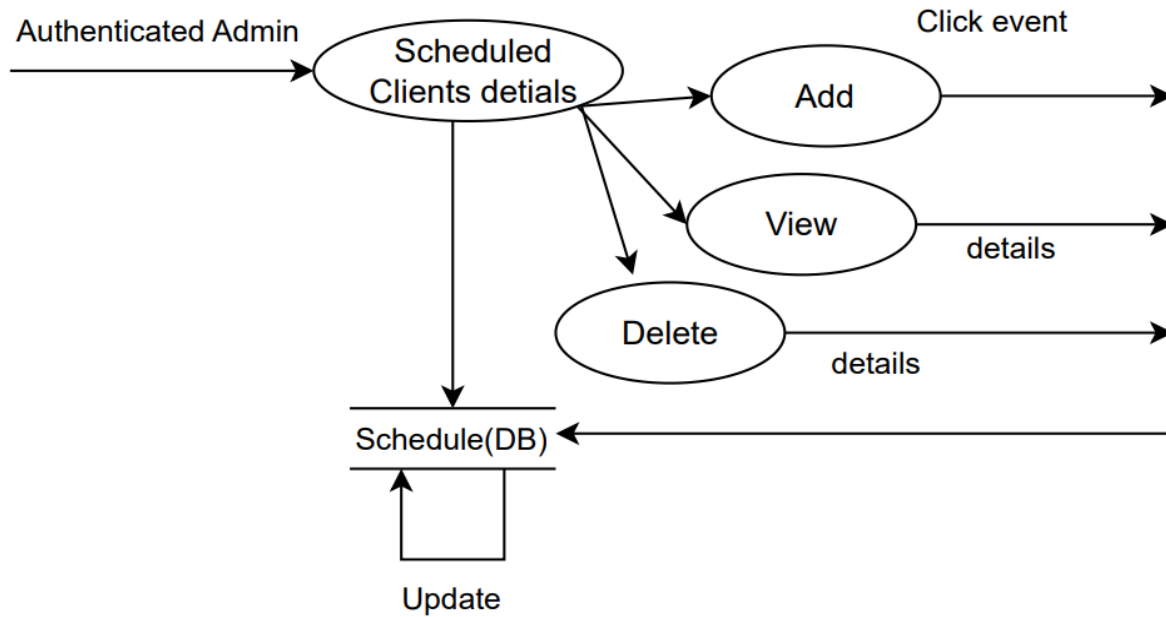


**First level Dataflow diagram (1 (b))****Second level Dataflow diagram (2a (i) )**

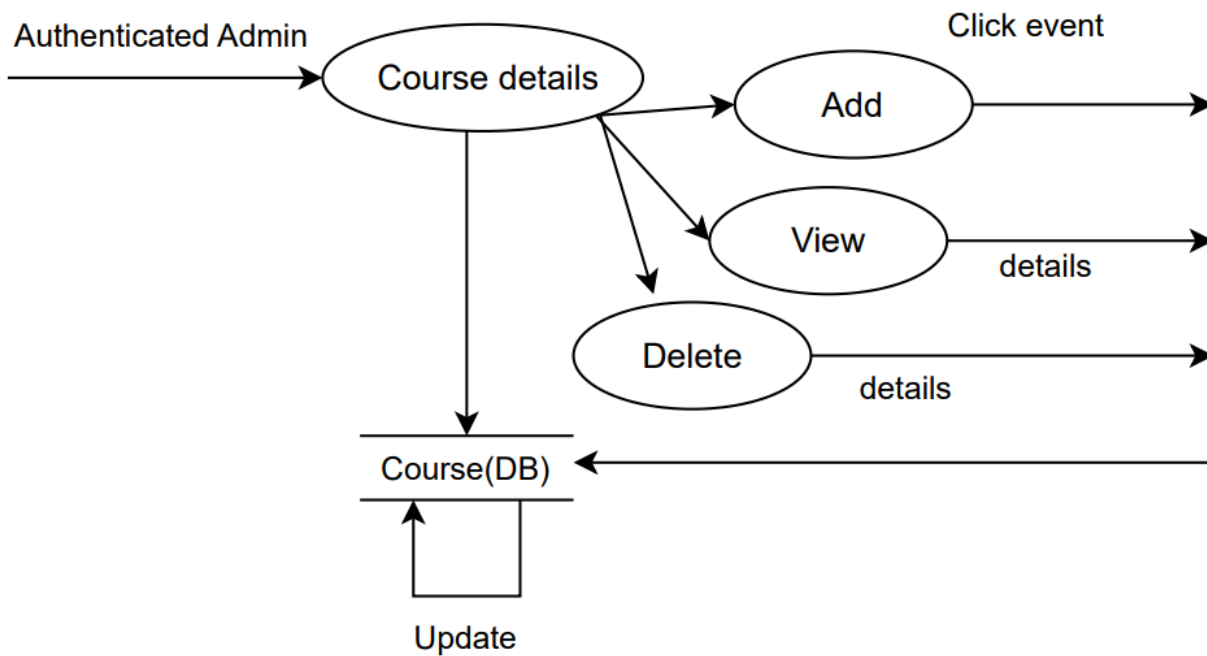
**Second level Dataflow diagram (2a (ii) )****Second level Dataflow diagram (2a (iii) )**

**Second level Dataflow diagram (2a (iv))****Second level Dataflow diagram (2b (i))**

Second level Dataflow diagram (2b (ii))

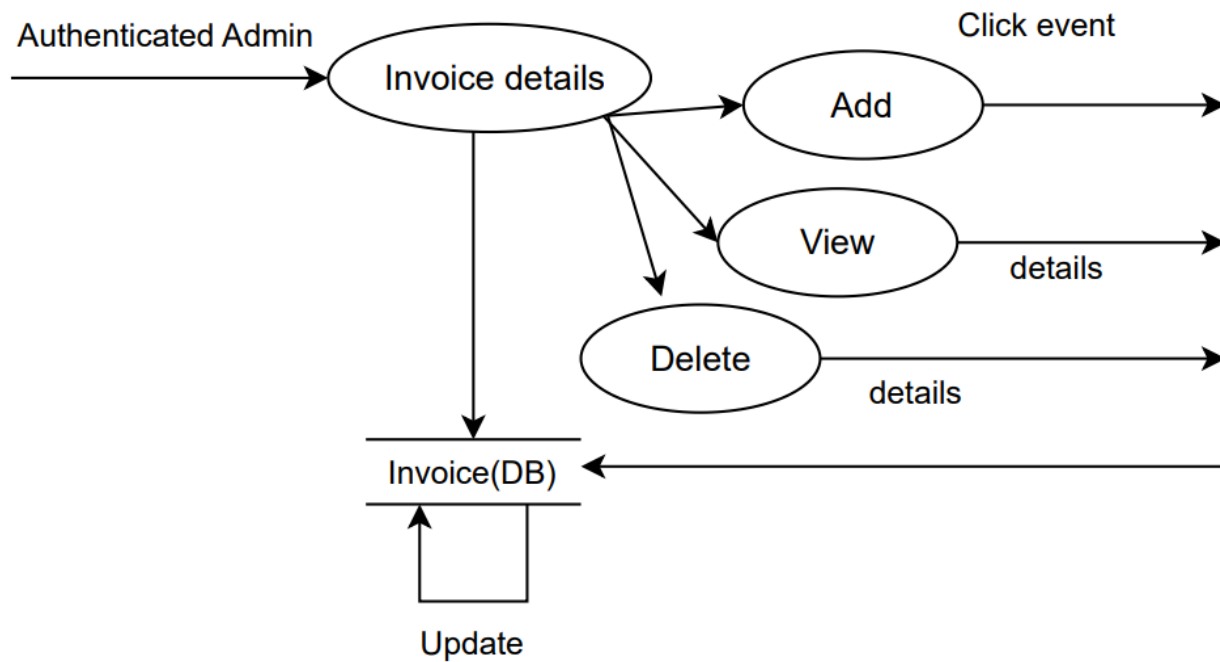


Second level Dataflow diagram (2b (iii))

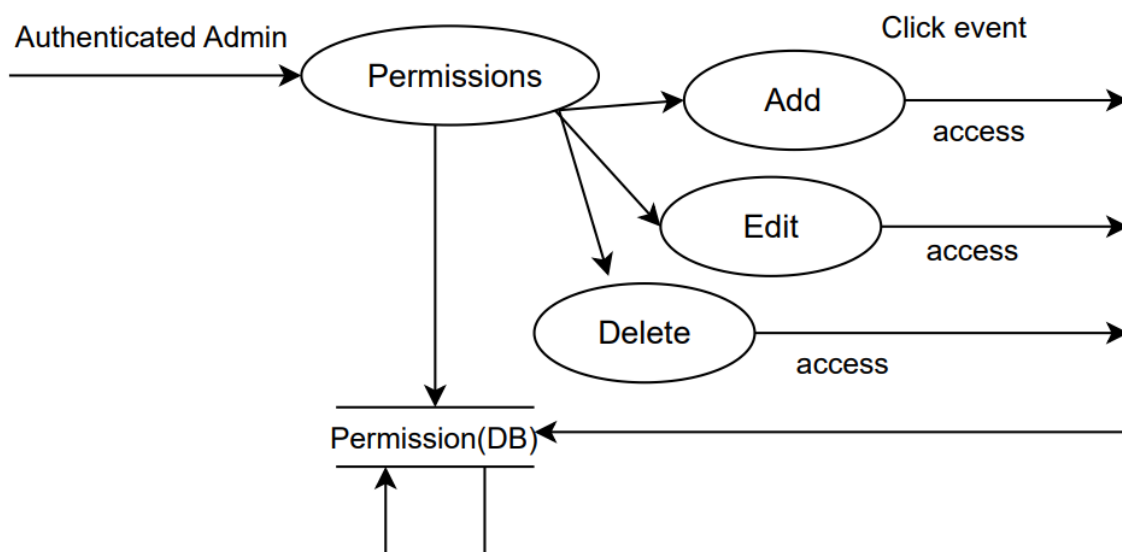




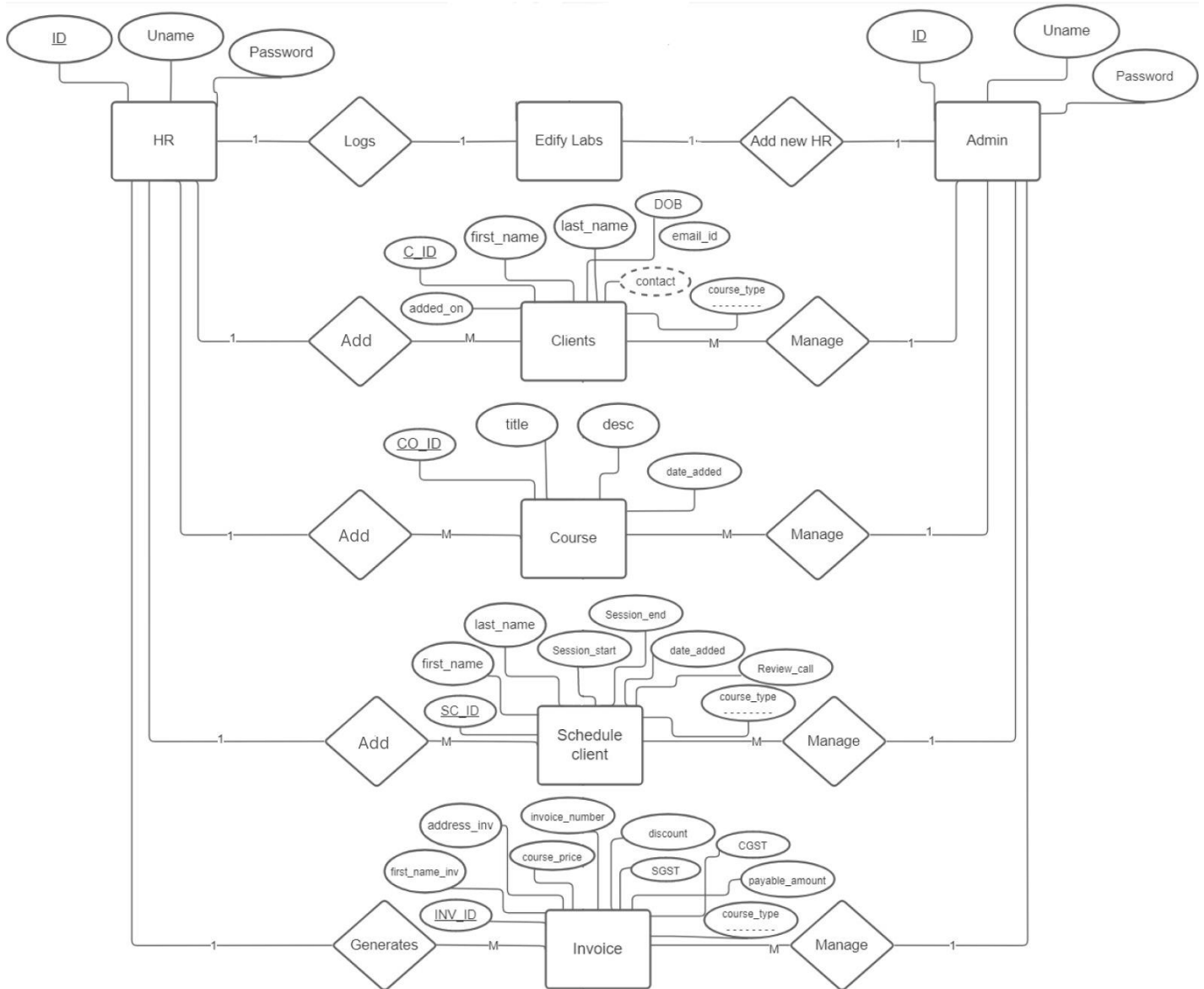
Second level Dataflow diagram (2b (iv))



Second level Dataflow diagram (2b (v))



### 3.3 Entity Relationship Diagram (ERD)



### 3.4 Table Structure

Table Name: clientsapp\_course

Sr.No	Field Name	Data Type	Constraints	Description
1	id	INTEGER	PRIMARY KEY AUTOINCREMENT	Identifier for the course
2	title	VARCHAR(100)	NOT NULL	Title of the course
3	desc	VARCHAR(100)	NOT NULL	Description of the course
4	date	DATETIME	NOT NULL	Date of the course

Table Name: clientsapp\_clients

Sr.No	Field Name	Data Type	Constraints	Description
1	id	INTEGER	PRIMARY KEY AUTOINCREMENT	Identifier for the client
2	first_name	VARCHAR(100)	NOT NULL	First name of the client
3	last_name	VARCHAR(100)	NOT NULL	Last name of the client
4	gender	VARCHAR(1)	NOT NULL	Gender of the client
5	company_name	VARCHAR(100)	NOT NULL	Company name of the client
6	email_id	VARCHAR(100)	NOT NULL	Email ID of the client
7	contact	VARCHAR(13)	NOT NULL	Contact number

				of the client
8	date_of_birth	VARCHAR(100)	NOT NULL	Date of birth of the client
9	address	TEXT	NOT NULL	Address of the client
10	date_added	DATETIME	NOT NULL	Date and time when the client was added
11	coaching_type_course_id	BIGINT	NOT NULL	Foreign key reference to the course

Table Name: scheduler\_scheduleclients

Sr.No	Field Name	Data Type	Constraints	Description
1	id	INTEGER	PRIMARY KEY AUTOINCREMENT	Identifier for the scheduled client
2	first_name_S	VARCHAR(100)	NOT NULL	First name of the client
3	last_name_S	VARCHAR(100)	NOT NULL	Last name of the client
4	review_call	BOOL	NOT NULL	Indicates if a review call is required
5	sessions	VARCHAR(100)	NOT NULL	Sessions for the scheduled client
6	my_time_field	TIME	NOT NULL	Time field for

				the schedule
7	date_added	DATETIME	NOT NULL	Date and time when the schedule was added
8	coaching_type_course_id	BIGINT	NOT NULL	Foreign key reference to the course
9	client_count	INTEGER unsigned	NOT NULL CHECK (client_count >= 0)	Number of clients

Table Name: invoice\_invoice

Sr.No	Field Name	Data Type	Constraints	Description
1	id	INTEGER	PRIMARY KEY AUTOINCREMENT	Identifier for the scheduled invoice
2	full_name_invoice	VARCHAR(100)	NOT NULL	Full name for invoice
3	address_invoice	VARCHAR(100)	NOT NULL	Address for invoice
4	GST_invoice	BOOL	NOT NULL	GST for invoice
5	invoice_number	VARCHAR(100)	NOT NULL	Invoice number
6	amount_invoice	DECIMAL	NOT NULL	Amount for the

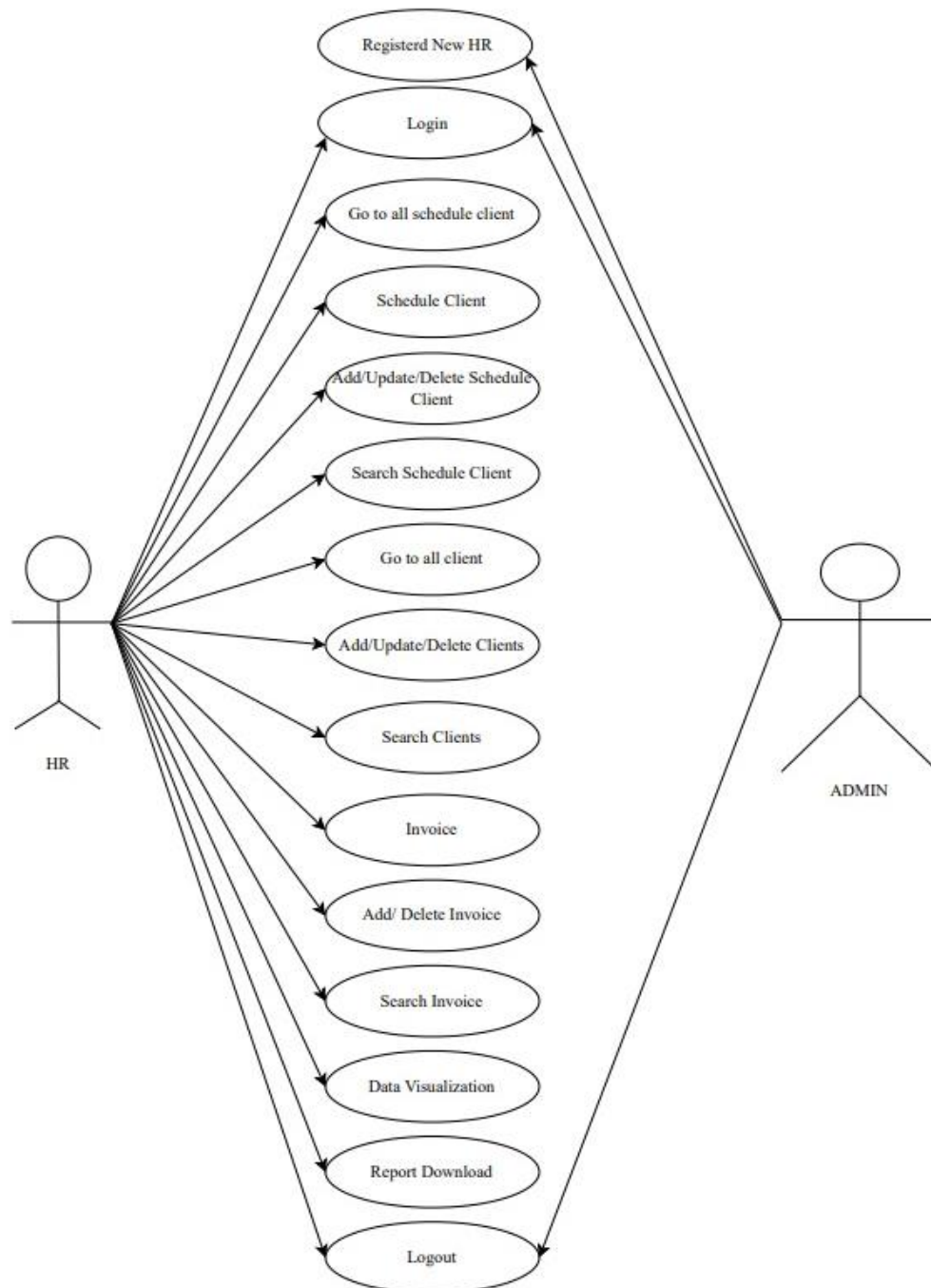
				invoice
7	date_added_invoice	DATE	NOT NULL	Date and time when the invoice was added
8	any_discount_invoice	DECIMAL	NOT NULL	Any discount for the invoice
9	coaching_type_course_ _invoice_id	BIGINT	NOT NULL	Foreign key reference to the course
10	cgst	INTEGER unsigned	NOT NULL CHECK (cgst >= 0)	CGST amount
11	sgst	INTEGER unsigned	NOT NULL CHECK (sgst >= 0)	SGST amount
12	total_amount_invoice	DECIMAL	NOT NULL	Total amount for the invoice
13	cgst_amount	DECIMAL	NOT NULL	CGST amount in the invoice
14	sgst_amount	DECIMAL	NOT NULL	SGST amount in the invoice
15	total_amount_payable_invoice	DECIMAL	NOT NULL	Total amount

				payable for the invoice
16	particular_invoice	TEXT	NOT NULL	Particulars for the invoice

Table Name: auth\_user

Sr.No	Field Name	Data Type	Constraints	Description
1	id	INTEGER	PRIMARY KEY AUTOINCREMENT	Identifier for the user
2	username	VARCHAR(100)	NOT NULL UNIQUE	User's username
3	first_name	VARCHAR(100)	NOT NULL	User's first name
4	last_name	VARCHAR(100)	NOT NULL	User's last name
5	email	VARCHAR(254)	NOT NULL	User's email address
6	date_joined	TIME	NOT NULL	Date and time when the user joined
7	last_login	DATETIME	NOT NULL	Last login date and time
8	password	VARCHAR(128)	NOT NULL	User's password
9	is_superuser	BOOL	NOT NULL	Indicates if the user is a superuser

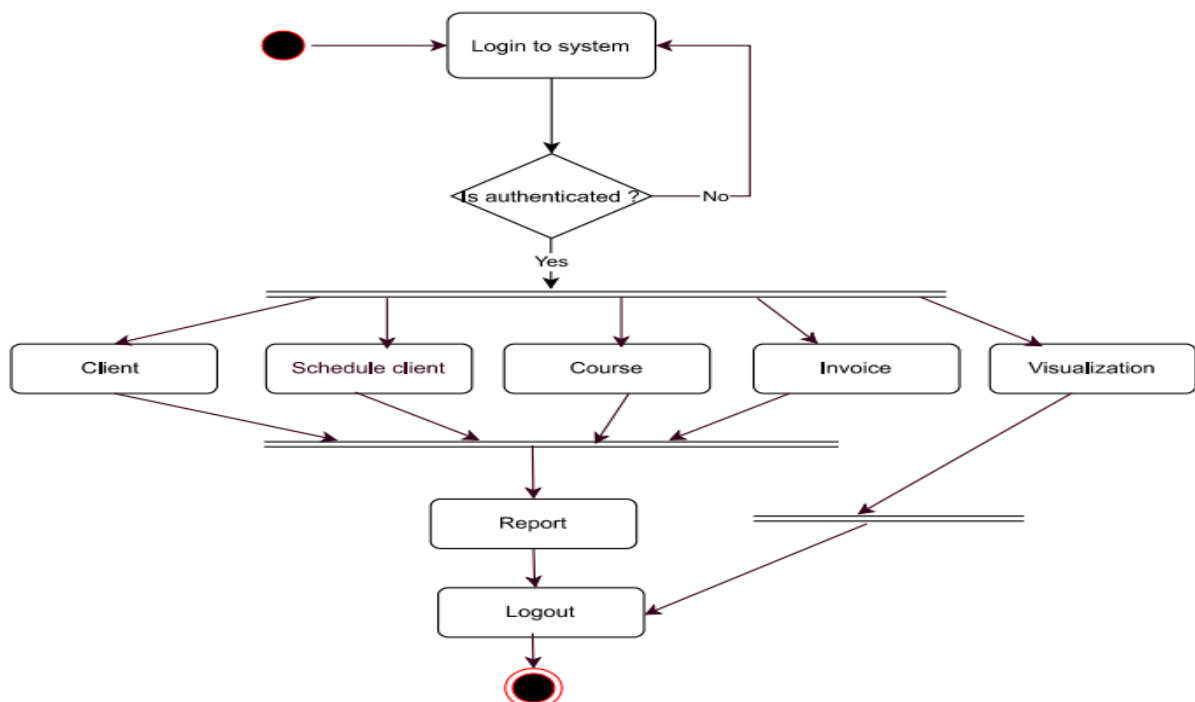
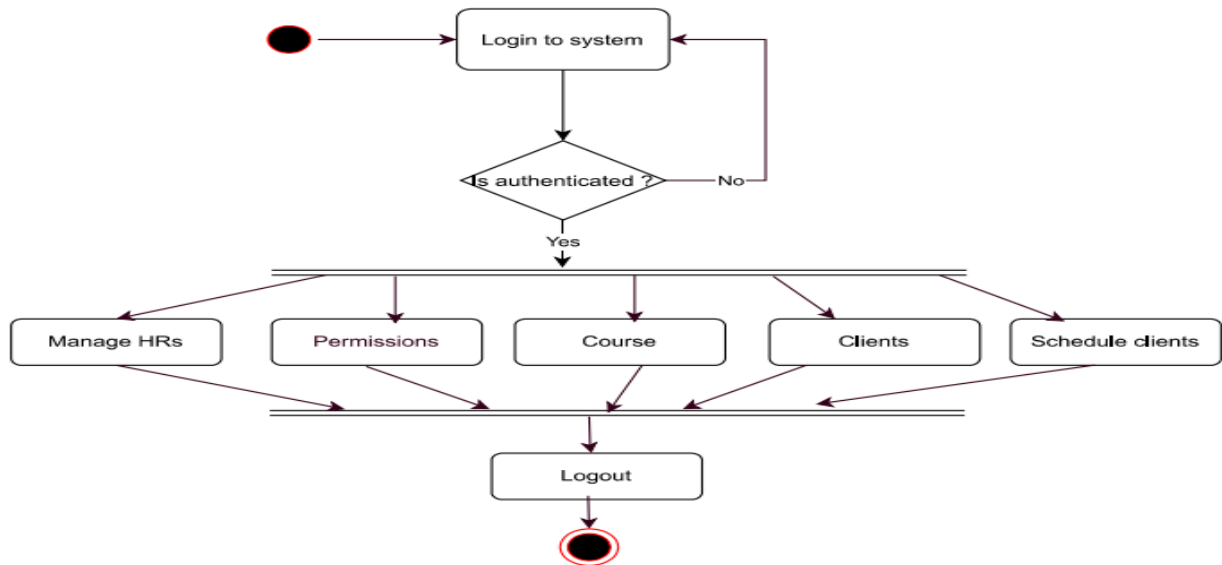
### 3.5 Use Case Diagrams



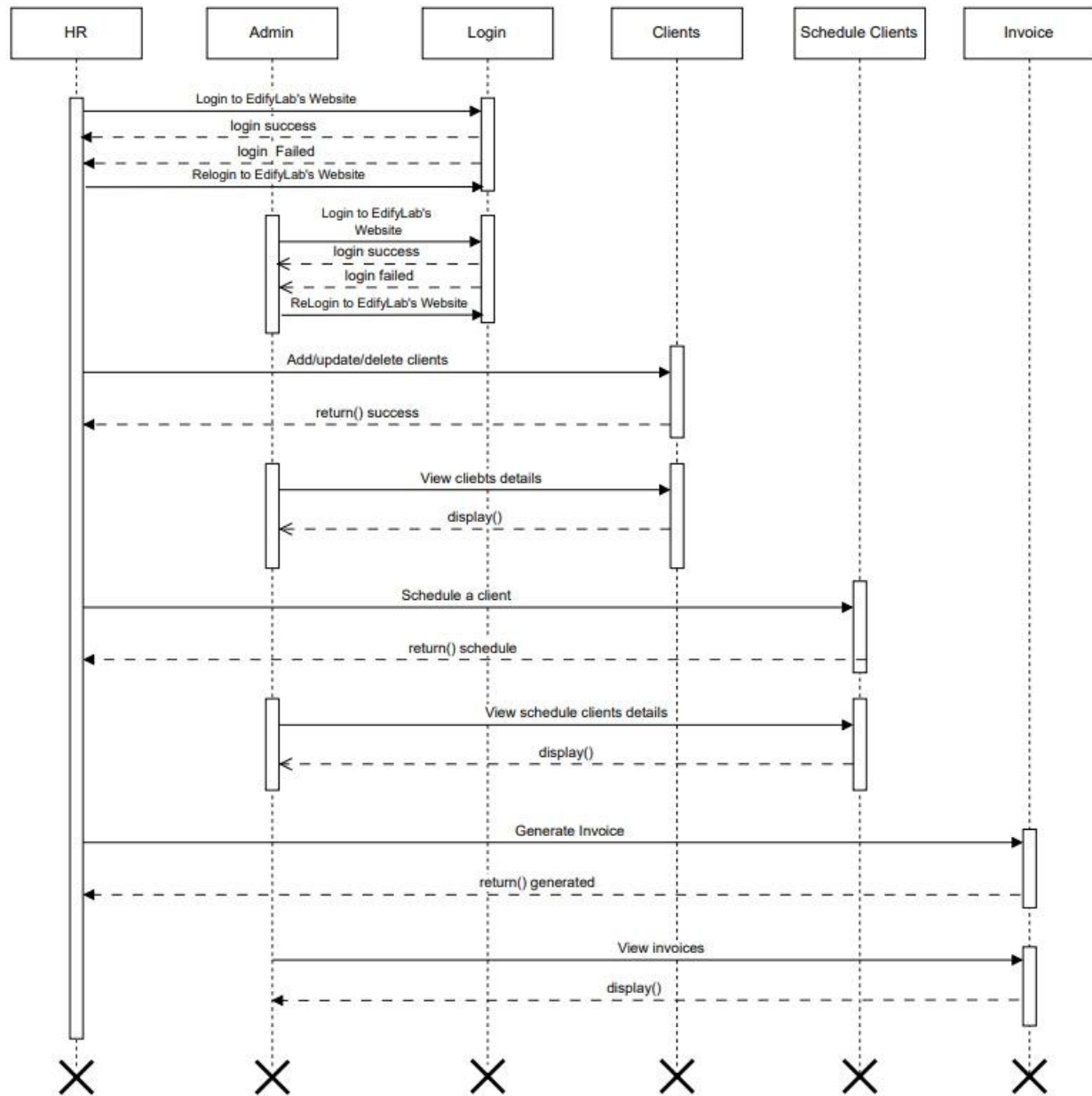


### 3.6 Activity Diagram

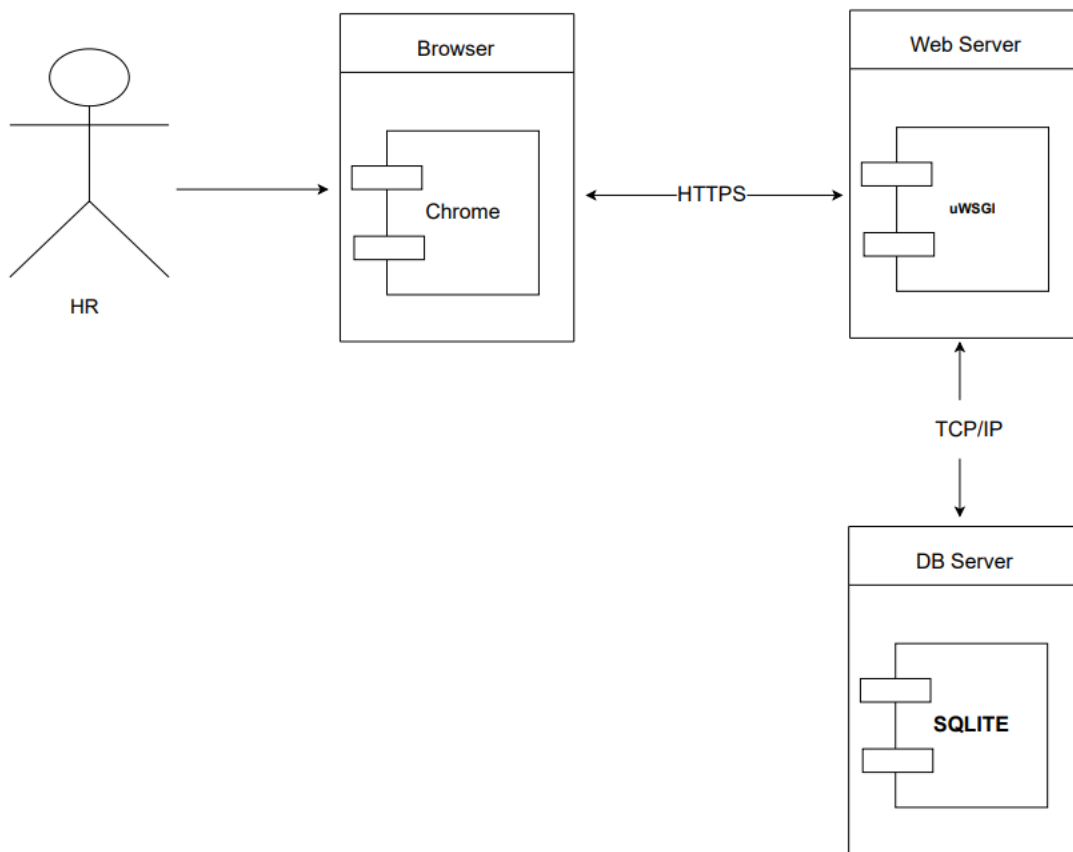
#### Admin Activity



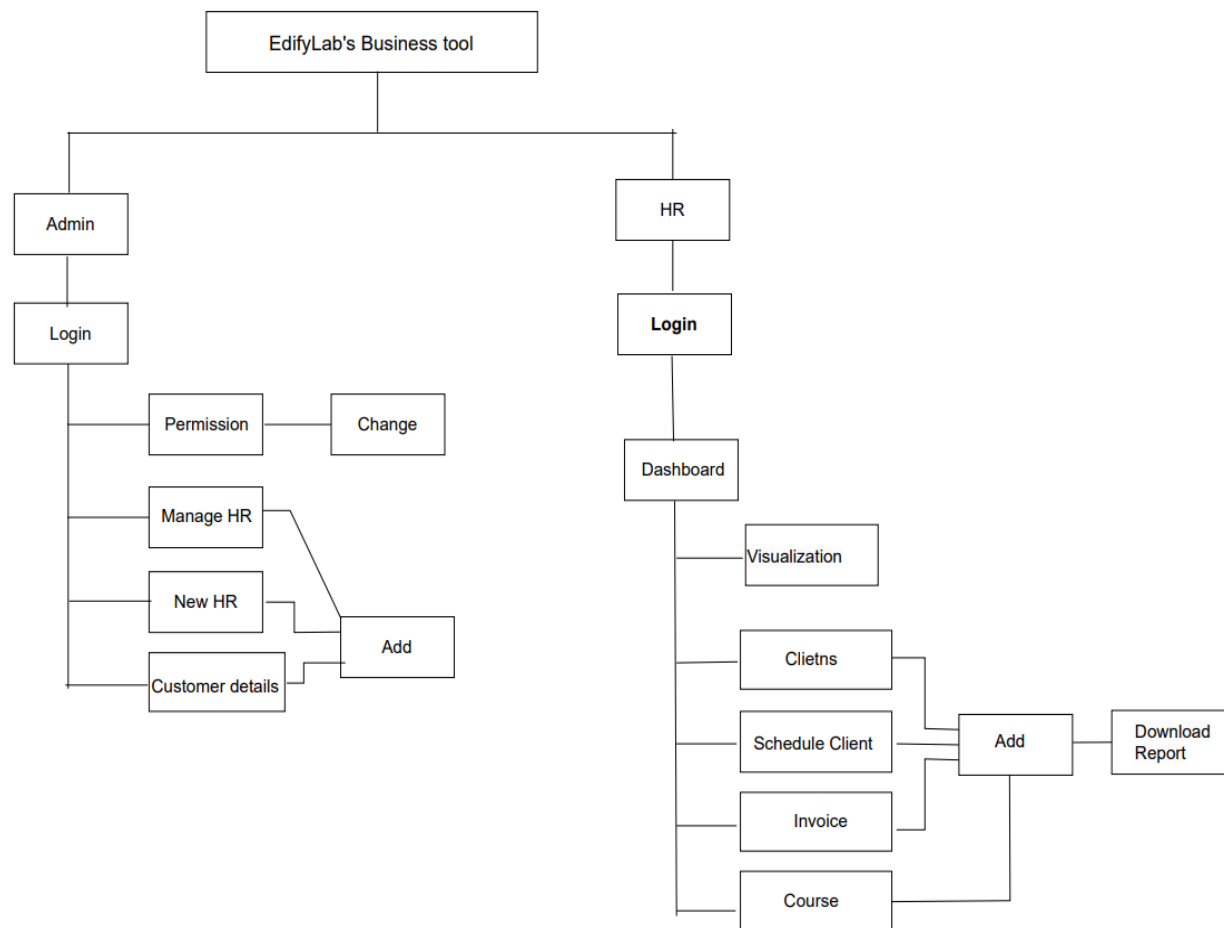
### 3.7 Sequence diagram



### 3.8 Deployment Diagram

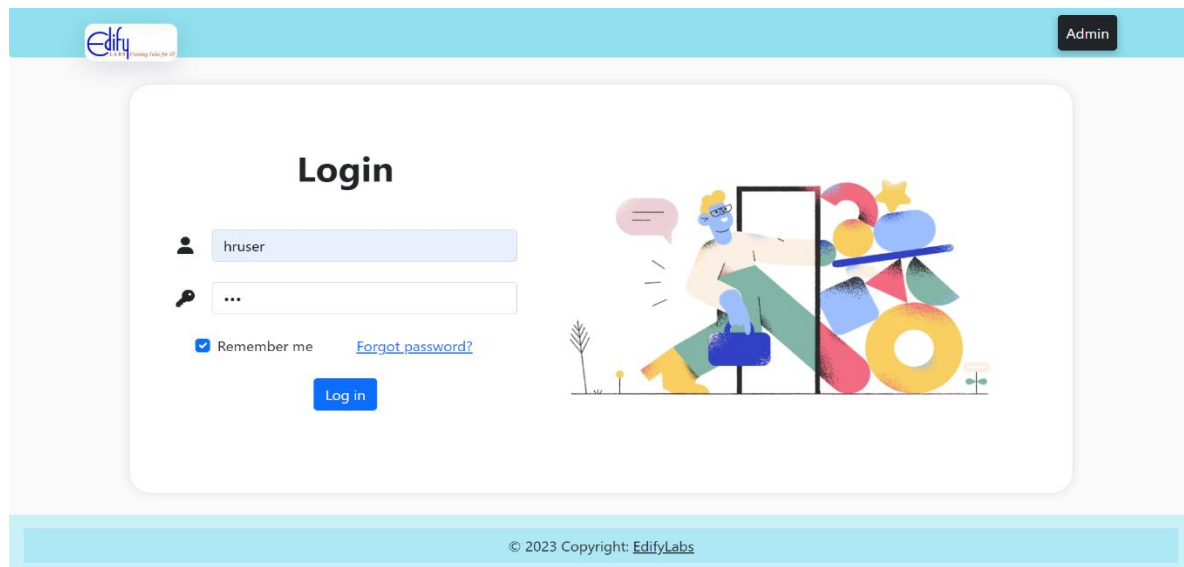


### 3.9 Component Diagram



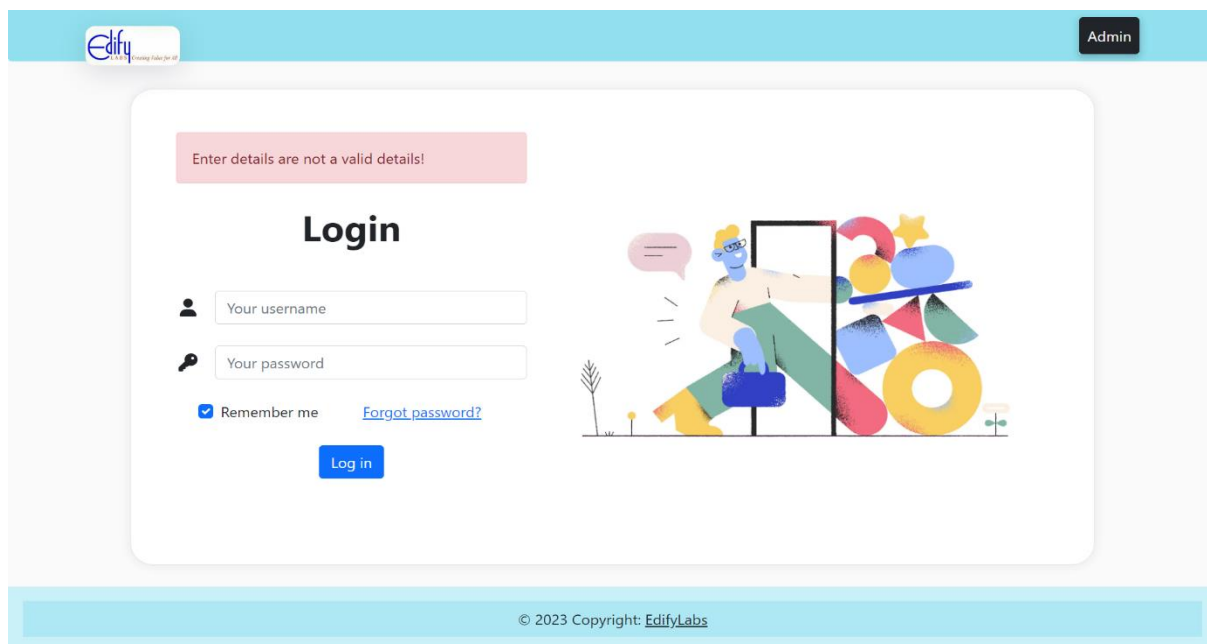
### 3.10 Sample Input and Output Screen

#### Input -> HR login screen (with invalid inputs)



The screenshot shows the EdifyLab HR login interface. At the top, there is a light blue header with the EdifyLab logo on the left and an 'Admin' button on the right. The main content area is a white box with a light gray border. On the left side of this box, the word 'Login' is centered at the top. Below it, there are two input fields: the first is labeled 'hruser' and the second is labeled '...'. Below the password field, there is a 'Remember me' checkbox (checked) and a link 'Forgot password?'. A blue 'Log in' button is at the bottom left of the login form. On the right side of the login form, there is a colorful illustration of a person sitting on a large, abstract shape, holding a briefcase, with a speech bubble above them. The footer of the page is a light blue bar with the text '© 2023 Copyright: EdifyLabs'.

#### Output -> Invalid message



The screenshot shows the EdifyLab HR login interface after an invalid login attempt. The layout is identical to the previous screenshot, but with an additional message. At the top of the white content box, there is a pink rectangular box with the text 'Enter details are not a valid details!'. The input fields now contain 'Your username' and 'Your password'. The 'Remember me' checkbox is checked, and the 'Forgot password?' link is still present. The blue 'Log in' button remains at the bottom left. The colorful illustration on the right is also present. The footer of the page is a light blue bar with the text '© 2023 Copyright: EdifyLabs'.

## Input -> Client's screen

id	First Name	Last Name	Email	Course	
CL-12	Vedika	Sawant	vedikaS@gmail.com	Personal coaching	🔄 + 🗑️ 📄
CL-11	KAVYA	CHAVALI	test@gmail.com	Personal coaching	🔄 + 🗑️ 📄
CL-10	Tanmay1	Barvi1	barvitanmay20@gmail.com	Personal coaching	🔄 + 🗑️ 📄
CL-9	appy	Barvi	appy0@gmail.com	Teenpreneur	🔄 + 🗑️ 📄
CL-8	test	test	test@gmail.com	LPP/MHL	🔄 + 🗑️ 📄
CL-7	Tanmay	Barvi	barvitanmay20@gmail.com	Personal coaching	🔄 + 🗑️ 📄
CL-5	Apurva	bomle	appy07@gmail.com	Personal coaching	🔄 + 🗑️ 📄
CL-4	Tanmay	Barvi	barvitanmay20@gmail.com	Personal coaching	🔄 + 🗑️ 📄

© 2023 Copyright: [EdifyLabs](#)

## Output -> Download Client details(particular)

**Details of - Vedika Sawant** June 21, 2023, 11:10 p.m.

**Personal Details**

Gender : Female      DOB : 11/08/1994      Address : Test , Raigad , 410222  
 Email : vedikaS@gmail.com      Contact : 9877362647      Company name : EdifyLabs


**Coaching Details**

Coaching type course : **Personal coaching**

**EdifyLabs**  
 Printed on : June 23, 2023, 9:07 p.m.  
 By : Vedika Sawant

Downloaded As PDF

## Input -> Scheduled Clients screen


INVOICE VISUALIZER

Hello , **Vedika Sawant** -> All Scheduled Clients Total scheduled clients : 12


Search already added Clients to schedule 🔍 🔊

These are the clients have been registered to Schedule a call. ✕

id	First Name	Last Name	Course	Session Start	Session Ends	Session no.	Review call			
SC-59	appy	zz	LPP/MHL	2023-05-18 23:49	6:52 a.m.	-	✓	👁	✎	🗑
SC-61	tanmay	bz	Personal coaching	2023-05-15 11:47	3:47 p.m.	1	✗	👁	✎	🗑
SC-63	tanmay	bz	Personal coaching	2023-05-16 22:20	2:20 a.m.	2	✗	👁	✎	🗑
SC-64	tanmay	bz	Personal coaching	2023-05-18 02:43	3:43 a.m.	3	✗	👁	✎	🗑
SC-66	Tanmay	Barvi	Personal coaching	2023-05-17 00:00	1 a.m.	1	✗	👁	✎	🗑
SC-67	Tanmay	Barvi	Personal coaching	2023-05-24 02:01	3:01 a.m.	2	✗	👁	✎	🗑
SC-68	Tanmay	Barvi	Personal coaching	2023-05-31 05:01	7:01 a.m.	3	✗	👁	✎	🗑
SC-69	Tanmay	Barvi	Personal coaching	2023-06-07 06:02	7:02 a.m.	4	✗	👁	✎	🗑
SC-71	apurva	bomle	Teenpreneur	2023-05-17 21:03	10:03 p.m.	-	✓	👁	✎	🗑
SC-72	appy	zz	LPP/MHL	2023-05-18 11:22	11:22 p.m.	-	✓	👁	✎	🗑
SC-73	tanmay	bz	Personal coaching	2023-05-20 10:00	11:53 a.m.	4	✗	👁	✎	🗑
SC-74	KAVYA	CHAVALI	Personal coaching	2023-06-08 23:38	1 a.m.	1	✗	👁	✎	🗑

© 2023 Copyright: EdifyLabs

## Output -> Download Scheduled Client details(particular)


INVOICE VISUALIZER

Details of your session May 17, 2023, 12:02 a.m.

Name : **Tanmay Barvi** Total Session(s) : 4


coaching type course : **Personal coaching**

1 - Session start - **2023-05-17 00:00** -> Session end - **01:00 AM**

2 - Session start - **2023-05-24 02:01** -> Session end - **03:01 AM**

3 - Session start - **2023-05-31 05:01** -> Session end - **07:01 AM**

4 - Session start - **2023-06-07 06:02** -> Session end - **07:02 AM**



Printed on : June 23, 2023, 9:13 p.m.  
By : Vedika Sawant

🔍 Downloaded As PDF





### 3.11 Reports

#### Output -> Download to excel (All clients)

	A	B	C	D	E	F	G	H	I	J	K
1	ID	First Name	Last Name	Email	Gender	DOB	Address	Contact	Company	Course	
2	CL-12	Vedika	Sawant	vedikaS@gmail.com	F	11/08/1995	Test , Raig	98773626	EdifyLabs	Personal coaching	
3	CL-11	KAVYA	CHAVALI	test@gmail.com	F	1999-11-1	SHIV SHIV,	919733226839		Personal coaching	
4	CL-10	Tanmay1	Barvi1	barvitanmay20@gmail.com	O	2023-06-0	Room no.: 9165837483			Personal coaching	
5	CL-9	appy	Barvi	appy0@gmail.com	O	2000-01-0	appy hom	9889353498		Teenpreneur	
6	CL-8	test	test	test@gmail.com	O	2023-06-0	test	34539453	xyz	LPP/MHL	
7	CL-7	Tanmay	Barvi	barvitanmay20@gmail.com		2023-05-3	Room no.: 23523525355			Personal coaching	
8	CL-5	Apurva	bomle	appy07@gmail.com	F	2004-01-1	Street: 10	9172697 8-		Personal coaching	
9	CL-4	Tanmay	Barvi	barvitanmay20@gmail.com	M	2003-10-2	Room no.: 93256963	TEST		Personal coaching	

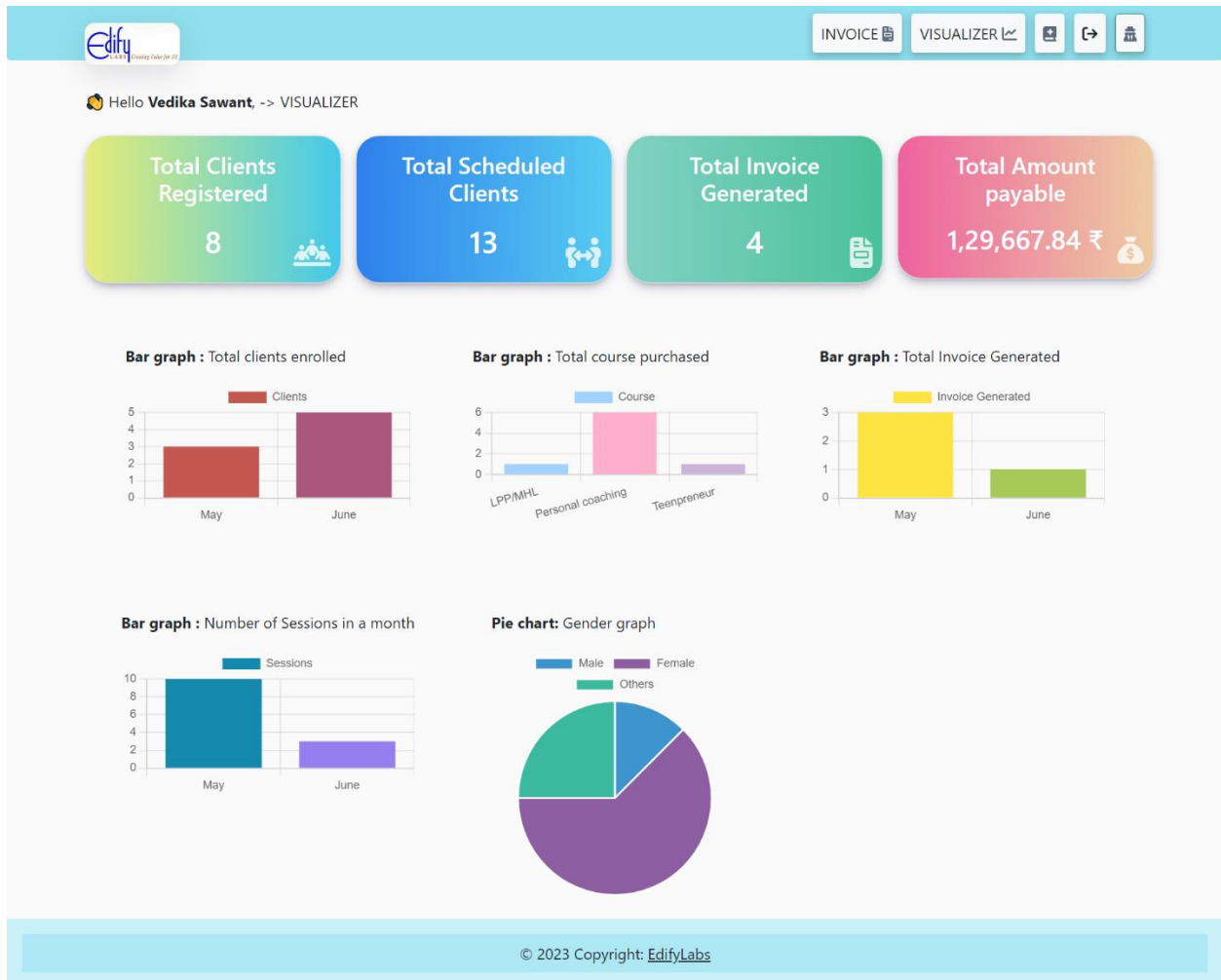
#### Output -> Download to excel (All Scheduled clients)

	A	B	C	D	E	F	G	H
1	ID	First Name	Last Name	Course	Review ca	Sessions start	Sessions end	count
2	CL-59	appy	zz	LPP/MHL	TRUE	2023-05-18 23:49	06:52:00	1
3	CL-61	tanmay	bz	Personal c	FALSE	2023-05-15 11:47	15:47:00	1
4	CL-63	tanmay	bz	Personal c	FALSE	2023-05-16 22:20	02:20:00	2
5	CL-64	tanmay	bz	Personal c	FALSE	2023-05-18 02:43	03:43:00	3
6	CL-66	Tanmay	Barvi	Personal c	FALSE	2023-05-17 00:00	01:00:00	1
7	CL-67	Tanmay	Barvi	Personal c	FALSE	2023-05-24 02:01	03:01:00	2
8	CL-68	Tanmay	Barvi	Personal c	FALSE	2023-05-31 05:01	07:01:00	3
9	CL-69	Tanmay	Barvi	Personal c	FALSE	2023-06-07 06:02	07:02:00	4
10	CL-71	apurva	bomle	Teenprene	TRUE	2023-05-17 21:03	22:03:00	1
11	CL-72	appy	zz	LPP/MHL	TRUE	2023-05-18 21:22	23:22:00	2
12	CL-73	tanmay	bz	Personal c	FALSE	2023-05-20 10:00	11:53:00	4
13	CL-74	KAVYA	CHAVALI	Personal c	FALSE	2023-06-08 23:38	01:00:00	1

#### Output -> Download to excel (All Scheduled clients)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	ID	Full Name	Address	GST no.	Invoice nu	Course	Particular	Amount	Discount	Total amo	CGST %	SGST %	CGST amo	SGST amo	Total amount payable	
2	INV-36	KAVYA RA' SHIV SHIV,	27AGSPC723/6/4		Personal c	Make It H		60000	20000	40000	9	9	3600	3600	47200	
3	INV-30	KAVYA RA' SHIV SHIV,	27AGSPC723/5/1		Personal c	kavya		60000	20000	40000	9	9	3600	3600	47200	
4	INV-33	tanamy1		23/5/3	Personal ct			60000	0	60000	9	9	5400	5400	70800	
5	INV-35	test		23/5/4	Personal ct			6888	0	6888	9	9	619.92	619.92	8127.84	

## Output -> Visualization charts reports



## Output -> Users status (Admin panel)

<input type="checkbox"/>	Username ↓	✕ Email address	First name	Last name	Staff status
<input type="checkbox"/>	admin	admin@gmail.com	Admin		✓
<input type="checkbox"/>	hruser	hr@gmail.com	Vedika Sawant		✓

2 users

# CHAPTER 4

# Coding

## 4.1 Algorithms

Algorithm for the login process in the Edify Labs HR Software System:

1. Start.
2. Declare variables `username` and `password`.
3. Read values of `username` and `password` from the user through the login form.
4. Validate that both `username` and `password` fields are not empty.
  - a. If either field is empty, display an error message and go to Step 9.
5. Query the database to find the user with the provided `username`.
  - a. If the user is not found in the database, display an error message, and go to Step 9.
6. Retrieve the hashed password associated with the found user.
7. Validate the provided `password` against the hashed password stored in the database.
  - a. If the validation fails, display an error message, and go to Step 9.
8. If all the fields are validated and the username is a valid user then show the Dashboard page , and go to Step 9
9. End.

Algorithm for the filtration in Edify Labs HR Software System:

The **filter algorithm** in Django involves applying conditions on a queryset to retrieve a subset of objects that match the specified criteria. The algorithm follows these steps:

- Import the necessary Django models for the data you want to filter.
- Define a function, let's call it `filter_data`, that takes in the `filter_criteria` as a parameter.
- Initialize a variable, let's call it `filtered_data`, with the initial queryset or model manager.  
For example, `filtered_data = Client.objects.all()`.
- Iterate over each attribute-value pair in the `filter_criteria`:  
Use the Django ORM's `filter()` method on `filtered_data` to apply the filter based on the attribute and value.  
For example, if the attribute is `'client_name'` and the value is `'John Doe'`, you would filter like this: `filtered_data = filtered_data.filter(client_name='John Doe')`.
- Return the `filtered_data`.

## 4.2 Code Snippets

```
""""
>> LOGIC FUNCTIONS FOR CLIENTS
""""

from django.shortcuts import render, redirect
from django.contrib.auth.decorators import login_required
from clientsapp.models import Clients
from .forms import ClientForm
from django.db.models import Q
import datetime
import openpyxl
from django.http import HttpResponse

# HOME -> FORM DISPLAY - ADD CLIENT - UPDATE CLIENT
@login_required(login_url='loginPage')
def home(request, id=0):
    if request.method == "GET":
        auth_user = request.user.first_name
        all_clients = Clients.objects.all()

        male_count = Clients.objects.filter(gender='M').count()
        female_count = Clients.objects.filter(gender='F').count()
        other_count = Clients.objects.filter(gender='O').count()
        print("Male count:", male_count)
        print(female_count)
        print(other_count)
```

```

gender_count = [male_count , female_count , other_count]
gender_name = ["Male" , "Female" , "Others"]

if id == 0:
    form = ClientForm()
else:
    client = Clients.objects.get(pk=id)
    form = ClientForm(instance=client)
    context = { "auth_user": auth_user, "page": "HOME | CLIENT | Edify Lab's Business Tool",
    "ClientForm": form, "total_clients": len(all_clients) , "gender_count" : gender_count ,
    "gender_name" : gender_name }
    return render(request, "home.html", context)
else:
    if id == 0:
        form = ClientForm(request.POST)
    else:
        client = Clients.objects.get(pk=id)
        form = ClientForm(request.POST, instance=client)
    if form.is_valid():
        form.save()

    return redirect('/clients-list/')

# DISPLAYS ALL CLIENTS
@login_required(login_url='loginPage')
def clients_list(request):
    all_clients = Clients.objects.all()
    page = "All clients | Edify Lab's Business Tool"
    if len(all_clients) != 0:
        context_clients_list = {
            "clients_list": all_clients, "total_clients": len(all_clients) , "page" : page}

```

```

else:
    error = "NO SILENTS 0"
    context_clients_list = {"error": error , "page":page}
    return render(request, "clients_template/clients_list.html/", context_clients_list)

```

#### # DELETE A SELECTED CLIENT

```

@login_required(login_url='loginPage')
def item_delete(request):
    if request.method == 'POST':
        item_ids = request.POST.getlist('item_ids')
        print(item_ids , "SELECTED")
        if item_ids:
            Clients.objects.filter(id__in=item_ids).delete()
    return redirect('/clients-list/')

```

#### # SEARCH CLIENT -> FNAME , LNAME , EMAIL , COURSE

```

@login_required(login_url='loginPage')
def search_client(request):
    if request.method == "POST":
        page = "Search clients | Edify Lab's Business Tool"
        title = request.POST["title"]

        lookups = Q(first_name__icontains=title) | Q(
            coaching_type_course__title__icontains=title) | Q(last_name__icontains=title) |
        Q(email_id__icontains=title)
        filter_title = Clients.objects.filter(lookups)

        print(len(filter_title), lookups)
        if title != "" and title is not None and len(filter_title) != 0:

            context_clients_list = {

```



```

        "clients_list": Clients.objects.all(), "filter_title": filter_title, "filter_title_total":
len(filter_title) , "page":page}
    else:
        error = "NO MATCH FOUND"
        context_clients_list = {"error": error , "page":page}
        return render(request, "clients_template/clients_list.html/", context_clients_list)
    else:
        print("-----SOMETHING WENT WRONG IN SEARCH CLIENT-----")
--")
        return redirect('/home/')

```

#### # DOWNLOAD TO PDF

```
@login_required(login_url='loginPage')
```

```
def download_client(request, id):
```

```
    if request.method == "GET":
```

```
        auth_user = request.user.first_name
```

```
        client = Clients.objects.get(pk=id)
```

```
        now = datetime.datetime.now
```

```
        context = {"auth_user": auth_user,
```

```
                    "page": f"EdifyLabs | download - {client.first_name}-{client.last_name}-{client.id}",
```

```
        "clients": client, "DT": now }
```

```
        return render(request, "clients_template/download_client.html", context)
```

#### # DOWNLOAD TO EXCEL

```
def download_clients_excel(request):
```

```
    clients = Clients.objects.all()
```

```
    workbook = openpyxl.Workbook()
```

```
    sheet = workbook.active
```

```
    header_row = ["ID", "First Name", "Last Name", "Email", "Gender", "DOB" , "Address",
```

```
"Contact" , "Company" ,"Course" ]
```

```
sheet.append(header_row)
```

```
for client in clients:
```

```
    data_row = [f"CL-{client.id}", client.first_name, client.last_name, client.email_id,
client.gender, client.date_of_birth,client.address , client.contact , client.company_name,
client.coaching_type_course.title]
```

```
    sheet.append(data_row)
```

```
response = HttpResponse(content_type='application/vnd.openxmlformats-officedocument.spreadsheetml.sheet')
```

```
response['Content-Disposition'] = 'attachment;
filename=clients_{datetime.datetime.now().strftime("%Y%m%d%H%M%S")}.xlsx'
```

```
workbook.save(response)
```

```
return response
```

```
"""
```

```
>> LOGIC FUNCTIONS FOR INVOICE GENERATION
```

```
-----
```

```
"""
```

```
from django.shortcuts import redirect, render
```

```
from .models import Invoice
```

```
from .forms import InvoiceForm
```

```
from django.contrib.auth.decorators import login_required
```

```
from num2words import num2words
```

```
from django.db.models import Q
```

```
import datetime
```

```
import openpyxl
```

```
from django.http import HttpResponseRedirect
```

```
# INVOICE DASHBOARD - DISPLAY FORM
```

```
@login_required(login_url='loginPage')
```

```
def invoice_dashboard(request):
```

```
    user = request.user.first_name
```

```
    form = InvoiceForm()
```

```
    all_invoice = Invoice.objects.all()
```

```
    highest_invoice = Invoice.objects.order_by('-invoice_number').first()
```

```
    print(highest_invoice)
```

```
    context = {"page": "INVOICE | Edify Lab's Business Tool ", "InvoiceForm" : form, "auth_user"
: user , "total_invoice": len(all_invoice)}
```

```
    return render(request, "invoice_template/invoice_dashboard.html" , context)
```

```
# DISPLAYS ALL THE INVOICES FROM THE DB
```

```
@login_required(login_url='loginPage')
```

```
def invoice_display(request):
```

```
    all_invoice = Invoice.objects.all()
```

```
    print(all_invoice)
```

```
    if len(all_invoice) != 0:
```

```
        context_invoice_list = { "page": "ALL INVOICE | Edify Lab's Business Tool ",
        "invoice_display": all_invoice, "total_invoice": len(all_invoice)}
```

```
    else:
```

```
        error = "NO INVOICE - 0"
```

```
        context_invoice_list = {"error": error , "page": " ALL INVOICE | Edify Lab's Business Tool
"}

```

```
    return render(request, "invoice_template/invoice_display.html/", context_invoice_list)
```

```
# ADD A NEW INVOICE TO THE DB FROM DASHBOARD
```

```
@login_required(login_url='loginPage')
```

```
def invoice_add(request):
```

```

if request.method == 'POST':
    form = InvoiceForm(request.POST)
    try:
        if form.is_valid():
            form.save()
            return redirect('invoice_display')
    except:
        print('Form is not valid')
else:
    form = InvoiceForm()
return render(request, 'invoice_template/invoice_dashboard.html', {'InvoiceForm': form})

```

#### # DELETE A SELECTED INVOICE

```

@login_required(login_url='loginPage')
def invoice_item_delete(request):
    if request.method == 'POST':
        item_ids = request.POST.getlist('item_ids_invoice')
        print(item_ids , "seleceted")
        if item_ids:
            Invoice.objects.filter(id__in=item_ids).delete()
    return redirect('invoice_display')

```

#### # GO TO DOWNLOAD PAGE INVOICE

```

@login_required(login_url='loginPage')
def invoice(request, id):
    auth_user = request.user.first_name
    invoice_of_id = Invoice.objects.get(pk=id)
    payable_amount = invoice_of_id.total_amount_payable_invoice
    payable_amount_text = num2words(payable_amount, lang='en').title() + ' Only'
    payable_amount_text = payable_amount_text.replace('-', ' ').replace(',', '')
    now = datetime.datetime.now

```

```

context = {"auth_user": auth_user,
           "title": f"EdifyLabs | invoice - {invoice_of_id.full_name_invoice}-
{invoice_of_id.invoice_number}", "invoice_of_id": invoice_of_id, "payable_amount_text":
payable_amount_text}

return render(request, "invoice_template/download_invoice.html", context)

# SEARCH INVOICE
@login_required(login_url='loginPage')
def search_invoice(request):
    if request.method == "POST":
        title = request.POST["title"]

        lookups = Q(full_name_invoice__icontains=title) | Q(
            coaching_type_course_invoice__title__icontains=title) |
Q(invoice_number__icontains=title) | Q(total_amount_payable_invoice__icontains=title)
        filter_title = Invoice.objects.filter(lookups)

        print(f" INVOICE {len(filter_title)} {lookups}")
        if title != "" and title is not None and len(filter_title) != 0:

            context_invoice_list = {
                "invoice_list_all": Invoice.objects.all(), "filter_title": filter_title, "filter_title_total":
len(filter_title)}
            else:
                error = "NO MATCH FOUND"
                context_invoice_list = {"error": error}
            return render(request, "invoice_template/invoice_display.html/", context_invoice_list)
        else:
            print("-----SOMETHING WENT WRONG IN SEARCH INVOICE-----
----")
            return redirect('/home/')

```

## # DOWNLOAD TO EXCEL

```
def download_invoice_excel(request):
```

```
    invoice_all = Invoice.objects.all()
```

```
    workbook = openpyxl.Workbook()
```

```
    sheet = workbook.active
```

```
    header_row = ["ID", "Full Name", "Address", "GST no.", "Invoice number", "Course" ,
"Particular detail", "Amount" , "Discount" , "Total amount" , "CGST %" , "SGST %" , "CGST
amount" ,"SGST amount" , "Total amount payable"]
```

```
    sheet.append(header_row)
```

```
    for inv in invoice_all:
```

```
        data_row = [f"INV-{inv.id}", inv.full_name_invoice, inv.address_invoice, inv.GST_invoice
,   inv.invoice_number   ,   str(inv.coaching_type_course_invoice),   inv.particular_invoice,
inv.amount_invoice,   inv.any_discount_invoice , inv.total_amount_invoice , inv.cgst , inv.sgst ,
inv.cgst_amount , inv.sgst_amount, inv.total_amount_payable_invoice ]
```

```
        sheet.append(data_row)
```

```
    response           =           HttpResponseRedirect(content_type='application/vnd.openxmlformats-
officedocument.spreadsheetml.sheet')
```

```
    response['Content-Disposition']           =           f'attachment;
filename=invoice_{datetime.datetime.now().strftime("%Y%m%d%H%M%S")}.xlsx'
```

```
    workbook.save(response)
```

```
    return response
```

## **CHAPTER 5**

# **Testing**

## 5.1 Test Strategy

1. **Objective:** The objective of the test strategy is to ensure the quality and reliability of the Edify Labs Business Tool by performing comprehensive testing across its modules. The strategy aims to identify and resolve any defects, validate the system's functionality, and verify that it meets the specified requirements.
2. **Testing Approach:** The testing approach will follow a combination of manual and automated testing techniques. Manual testing will be used for functional testing, user interface testing, and exploratory testing. Automated testing will be employed for regression testing, data-driven testing, and performance testing.
3. **Test Levels:** The testing will be conducted at the following levels:
  - a. **Unit Testing:** Individual units or components will be tested in isolation to ensure their correctness.
  - b. **Integration Testing:** The integration of various modules will be tested to ensure their proper functioning together.
  - c. **System Testing:** The entire system will be tested to validate its compliance with functional and non-functional requirements.
  - d. **Acceptance Testing:** The system will be tested by end-users or stakeholders to ensure it meets their expectations.
4. **Test Types:** The following test types will be performed:
  - a. **Functional Testing:** Validate the functionality of each module, including HR authentication, HR dashboard, scheduling, filtration, data visualization, report generation, invoice generation, and data downloading.
  - b. **User Interface Testing:** Verify the user interface elements such as forms, buttons, menus, and navigation for usability and consistency.
  - c. **Performance Testing:** Measure the system's performance under expected and peak loads to ensure it can handle the required user interactions.
  - d. **Security Testing:** Validate the system's security



measures, including user authentication, data encryption, and protection against common vulnerabilities. e. Regression Testing: Re-test previously implemented functionality after any changes or enhancements to ensure that existing features are not negatively affected. f. Compatibility Testing: Verify the compatibility of the system with different browsers, operating systems, and devices. g. Usability Testing: Evaluate the system's ease of use, intuitiveness, and overall user experience. h. Integration Testing: Test the integration of different modules and components to ensure proper data flow and functionality.

5. Test Data: Test data will be carefully selected to cover a wide range of scenarios, including valid and invalid inputs, boundary values, and edge cases. Realistic and representative data will be used to ensure the accuracy and effectiveness of the tests.
6. Test Environment: A dedicated test environment will be set up that closely resembles the production environment. This environment will include the necessary hardware, software, databases, and network configurations to support testing activities.
7. Test Deliverables: The following deliverables will be produced as part of the testing process:
  - a. Test Plan: A comprehensive document outlining the test strategy, objectives, approach, scope, and schedule.
  - b. Test Cases: Detailed test cases that cover all functional and non-functional requirements.
  - c. Test Data: Well-defined test data sets used for testing different scenarios.
  - d. Test Logs: Logs documenting the execution of test cases, including test results, issues encountered, and any additional observations or notes.
  - e. Defect Reports: Reports documenting any defects found during testing, including their severity, priority, and steps to reproduce.
8. Test Schedule: A detailed test schedule will be created, considering the project timeline, development milestones, and resource availability. The schedule will allocate sufficient time for each testing phase, including test planning, test case development, test execution, and defect management.
9. Risk and Issue Management: A risk assessment will be conducted to identify potential risks and issues that may impact the testing process. Mitigation strategies will be defined to minimize the impact of these risks, and regular communication channels will be established to report and address any issues that arise during testing.

10. Test Execution and Reporting: Test execution will be carried out according to the defined test plan. Test progress and results will be regularly reported to the project stakeholders. Test logs, defect reports, and other relevant documentation will be maintained and shared with the development team for issue resolution.
11. Test Completion Criteria: The following criteria will be used to determine when testing is considered complete: a. All planned test cases have been executed and passed. b. High-priority defects have been resolved or mitigated. c. Acceptance criteria defined by stakeholders have been met. d. Test coverage for critical functionality is satisfactory. e. Test exit criteria, as defined in the test plan, have been fulfilled.

## 5.2 Test Case/Test Script

### **#Test code for Login Authentication**

```
from django.test import TestCase, Client
from django.urls import reverse
from django.contrib.auth.models import User

class LoginPageTestCase(TestCase):
    def setUp(self):
        self.client = Client()
        self.signup_url = reverse('signup')
        self.login_url = reverse('loginPage')
        self.logout_url = reverse('logoutPage')
        self.ourstory_url = reverse('ourstory')
        self.user = User.objects.create_user(username='testuser', password='testpassword')

    def test_signup(self):
        response = self.client.get(self.signup_url)
        self.assertEqual(response.status_code, 200)

        data = {
            'name': 'testuser',
            'email': 'testuser@example.com',
            'pwd': 'testpassword',
            'pwdfinal': 'testpassword'
        }
```

```
response = self.client.post(self.signup_url, data)
self.assertEqual(response.status_code, 200)
self.assertContains(response, 'LOGIN')

def test_login_page(self):
    response = self.client.get(self.login_url)
    self.assertEqual(response.status_code, 200)

    data = {
        'login_username': 'testuser',
        'login_pass': 'testpassword'
    }
    response = self.client.post(self.login_url, data)
    self.assertEqual(response.status_code, 302)
    self.assertRedirects(response, reverse('home'))

    # Attempt login with incorrect credentials
    data = {
        'login_username': 'testuser',
        'login_pass': 'incorrectpassword'
    }
    response = self.client.post(self.login_url, data)
    self.assertEqual(response.status_code, 200)
    self.assertContains(response, 'Error_message_unauthorized')

def test_logout_page(self):
    self.client.login(username='testuser', password='testpassword')
    response = self.client.get(self.logout_url)
    self.assertEqual(response.status_code, 302)
    self.assertRedirects(response, reverse('loginPage'))
```

```
def test_ourstory_page(self):
    response = self.client.get(self.ourstory_url)
    self.assertEqual(response.status_code, 200)

def test_authenticated_home_page(self):
    self.client.login(username='testuser', password='testpassword')
    response = self.client.get(reverse('home'))
    self.assertEqual(response.status_code, 200)
    self.assertContains(response, 'OkNoted | HOME')

def test_unauthenticated_home_page(self):
    response = self.client.get(reverse('home'))
    self.assertEqual(response.status_code, 302)
    self.assertRedirects(response, reverse('loginPage'))
```

```
class SignupPageTestCase(TestCase):
    def setUp(self):
        self.client = Client()
        self.signup_url = reverse('signup')

    def test_signup_page(self):
        response = self.client.get(self.signup_url)
        self.assertEqual(response.status_code, 200)
```

**#Test code for Client's task**

```
from django.test import TestCase
from django.urls import reverse
from clientsapp.models import Clients

class ClientsAppTestCase(TestCase):

    def setUp(self):
        self.client1 = Clients.objects.create(
            first_name='Tanmay',
            last_name='Barvi',
            email_id='Tanmay.Barvi@example.com',
            gender='M',
            date_of_birth='2000-08-13',
            address='xyz test',
            contact='9325949697',
            company_name='ABC Company',
            coaching_type_course='Personal coaching'
        )

    def test_home_view(self):
        response = self.client.get(reverse('home'))
        self.assertEqual(response.status_code, 200)
        self.assertTemplateUsed(response, 'home.html')

    def test_clients_list_view(self):
```

```
response = self.client.get(reverse('clients_list'))
self.assertEqual(response.status_code, 200)
self.assertTemplateUsed(response, 'clients_template/clients_list.html/')

def test_item_delete_view(self):
    # Prepare data for testing deletion
    item_ids = [self.client1.id]
    initial_count = Clients.objects.count()

    response = self.client.post(reverse('item_delete'), {'item_ids': item_ids})
    self.assertEqual(response.status_code, 302) # Redirect status code
    self.assertEqual(Clients.objects.count(), initial_count - 1)

def test_search_client_view(self):
    # Prepare data for testing search
    search_term = 'Tanmay'

    response = self.client.post(reverse('search_client'), {'title': search_term})
    self.assertEqual(response.status_code, 200)
    self.assertTemplateUsed(response, 'clients_template/clients_list.html/')

def test_download_client_view(self):
    client_id = self.client1.id

    response = self.client.get(reverse('download_client', args=[client_id]))
    self.assertEqual(response.status_code, 200)
    self.assertTemplateUsed(response, 'clients_template/download_client.html')

def test_download_clients_excel_view(self):
    response = self.client.get(reverse('download_clients_excel'))
    self.assertEqual(response.status_code, 200)
```

```
self.assertEqual(response['Content-Type'], 'application/vnd.openxmlformats-officedocument.spreadsheetml.sheet')
```

## 5.3 Defect Report/Test Log

---

### Test Log

---

Date: 2023-06-24

Time: 09:30 AM

Environment: Production

---

Test Case 1: Signup - Successful

---

Result: Pass

Duration: 0.5s

User successfully signed up with valid credentials.

---

Test Case 2: Signup - Error (Missing fields)

---

Result: Pass

Duration: 0.3s

Validation error triggered for missing required fields.



-----  
Test Case 3: Signup - Error (Username already taken)  
-----

Result: Pass

Duration: 0.4s

Validation error triggered for an already taken username.

-----  
Test Case 4: Login - Successful  
-----

Result: Pass

Duration: 0.6s

User successfully logged in with valid credentials.

-----  
Test Case 5: Login - Error (Invalid credentials)  
-----

Result: Pass

Duration: 0.5s

Authentication error triggered for invalid credentials.

-----  
Test Case 6: Logout  
-----

Result: Pass

Duration: 0.2s

User successfully logged out.

=====

Test Summary

=====

Total Test Cases: 6

Passed: 6

Failed: 0

Errors: 0

Skipped: 0

Total Duration: 3.2s

All tests passed without any errors or failures.

## **CHAPTER 6**

# **Limitations of Proposed System**

1. Reliance on internet connectivity, as it is a web-based application. If a user is unable to connect to the internet, they may not be able to access the system or perform certain tasks.
2. Integration with other software systems: Edify Labs may not have seamless integration capabilities with other HR or business software systems, which could limit its overall usefulness for companies that rely on multiple tools.
3. While Edify Labs Business Tool aims to provide a more personalized and tailored HR experience, it may not be suitable for larger organizations with complex HR needs, as the system is designed with a focus on smaller-scale operations.
4. Learning curve for new users: The tool may have a learning curve for users who are unfamiliar with the system or have limited technical expertise. Adequate training and documentation should be provided to help users effectively navigate and utilize the tool's features.

## **CHAPTER 7**

# **Proposed Enhancements**

1. Mobile application: Currently, Edify Labs Business Tool is a web-based application. A mobile application could be developed to allow HR professionals to manage client data on the go.
2. Advanced analytics: While Edify Labs Business Tool offers data visualization and reporting features, incorporating advanced analytics tools like predictive analytics and machine learning could help HR professionals identify trends and patterns in employee data, enabling them to make more informed decisions.
3. Integration with video conferencing: The system could be enhanced to allow for video conferencing with employees, providing a more personalized and interactive experience.
4. Customization: The system could be enhanced to allow for greater customization and personalization, providing HR professionals.
5. Employee engagement and feedback: Implement features to measure and enhance client engagement, such as surveys, feedback mechanisms, and recognition programs. This can help HR professionals understand client's satisfaction and take proactive steps to improve engagement and overall process.

## **CHAPTER 8**

# **Conclusion**

In conclusion, the Edify Labs project has been a significant endeavor aimed at developing a comprehensive and user-friendly business tool for HR professionals. Throughout the project, I have successfully created a web-based application that simplifies and streamlines various HR processes, including client management, data analysis, and reporting.

The Edify Labs Business Tool offers numerous benefits, such as centralized data management, intuitive user interfaces, and data visualization capabilities, enabling HR professionals to make informed decisions and optimize their HR operations. By automating routine tasks, the tool helps save time and reduce administrative burdens, allowing HR teams to focus on strategic initiatives and employee engagement.

During the development process, I implemented key features such as client management, search functionality, data export to Excel, and user authentication. I also ensured the security of sensitive information through password hashing and user authentication mechanisms.

While the project has achieved significant milestones, there are opportunities for future enhancements. These include the development of a mobile application, incorporation of advanced analytics, integration with video conferencing, and customization options to cater to diverse HR requirements.

I am confident that the Edify Labs Business Tool will empower HR professionals, streamline their workflows, and contribute to the success of organizations in managing their human resources effectively.



## **CHAPTER 9**

# **Bibliography**

### Django Documentation:

Official Django documentation: <https://docs.djangoproject.com/>

### Python Documentation:

Official Python documentation: <https://docs.python.org/>

### Web Development Resources:

Mozilla Developer Network (MDN) Web Docs: <https://developer.mozilla.org/en-US/docs/Web>

W3Schools: <https://www.w3schools.com/>

Stack Overflow: <https://stackoverflow.com/>

### Books:

"Django for Beginners" by William S. Vincent


### Online Tutorials and Courses:

Django for Beginners: <https://djangoforbeginners.com/>


# **CHAPTER 10**


# **User Manual**

**Login Page** - From here HR can login


Get Started

## Login






☒ Remember me [Forgot password?](#)



© 2023 Copyright: [EdifyLabs](#)

**HR Dashboard** – Here new clients can be added and redirected to various pages

INVOICE VISUALIZER

Hello **Vedika Sawant**, -> CLIENT

Total clients : 8


Add a new client here

First name

Last name

Gender

Date of birth




Address

Email

Contact


Company Name

Coaching Type



© 2023 Copyright: [EdifyLabs](#)

**Schedule page** – Here HR Schedules a call or a meeting if a client is already added


INVOICE VISUALIZER

Hello , **Vedika Sawant** , -> Schedule a Client

Search already added Clients to schedule

First name

Last name

Review call  
☐

Coaching Type  
Select type of coaching

Session Start  
dd-mm-yyyy --:--


Session End  
--:--

Schedule client

id	First Name	Last Name	Email	Course	Added on
CL-5	Apurva	bomle	appy07@gmail.com	Personal coaching	May 31, 2023, 10:44 p.m.

© 2023 Copyright: [EdifyLabs](#)

**Invoice page** – Here HR generates invoices


INVOICE VISUALIZER

Hello **Vedika Sawant** , -> INVOICE Total invoices added till now : 4

Generate and save Invoice from here

Full name  
e.g: Full name

Address  
e.g: xyz ,Maharashtra , India

GST NO  
e.g: 27AGSPC7785J1ZM

Coaching Type  
Select type of coaching

Particular details  
e.g: xyz for 4 Weeks

Amount  
e.g: 10000.00

Discount (if any)  
0

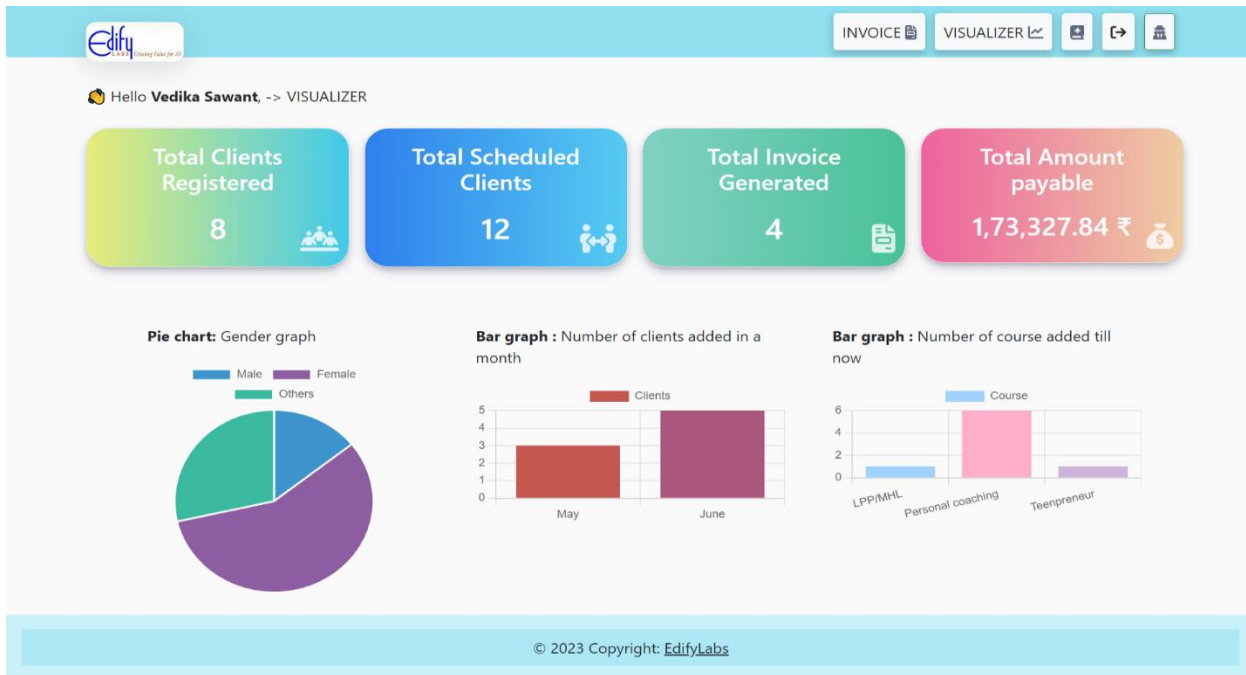
CGST %  
9

SGST %  
9

Save Invoice

© 2023 Copyright: [EdifyLabs](#)

## Visualization page – Here HR can see the analysis



## Add course – Here HR can add new courses

**Courses** | [Home](#) > [Clientsapp](#) > [Courses](#) + Add course

----- Go 0 of 3 selected

<input type="checkbox"/>	Title	Small Message?	Date
<input type="checkbox"/>	Teenpreneur	teen	May 2, 2023, 9:26 p.m.
<input type="checkbox"/>	LPP/MHL	msg lpp	May 2, 2023, 9:25 p.m.
<input type="checkbox"/>	Personal coaching	Personal	May 2, 2023, 9:25 p.m.

3 courses

Copyright © 2023 . All rights reserved. Jazzmin version 2.6.0 - Tanmay

## Visualization page – Admin login page



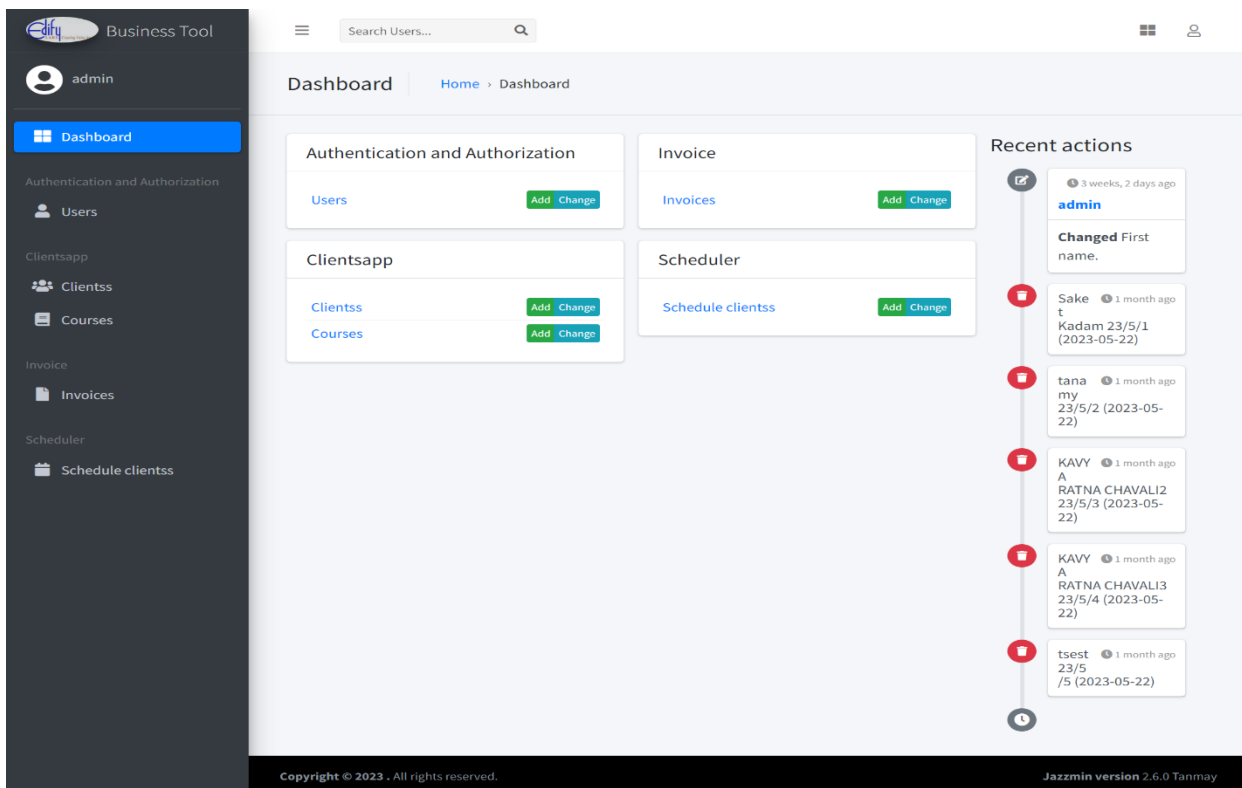
Welcome

Username

Password

[Log in](#)

## Admins page – Here Admin manages all the system



The screenshot displays the Admin Dashboard of the EdifyLab Business Tool. The interface includes a dark sidebar with navigation links for Dashboard, Users, Clientsapp, Invoices, and Scheduler. The main content area is titled 'Dashboard' and contains several management cards: 'Authentication and Authorization' (Users), 'Invoice' (Invoices), 'Clientsapp' (Clientss, Courses), and 'Scheduler' (Schedule clientss). Each card has 'Add' and 'Change' buttons. On the right, a 'Recent actions' timeline shows activities like 'Changed First name' and 'Sake t Kadam' by the admin. The footer indicates 'Copyright © 2023 . All rights reserved.' and 'Jazzmin version 2.6.0 Tanmay'.