**FitGenius**

**Darshan Jain**

**MCA**

**(Full Stack Development)**

**Astha Agrawal**

**MCA**

**(Full Stack Development)**

****

**FitGenius**

Thesis submitted in partial fulfilment

of the requirements of the degree of

**MCA**

**(Full Stack Development)**

**Master of Computer Applications**

By

**Darshan Jain (2023-M-11081998)**

**Astha Agrawal(2023-M-30082002)**

Under the Supervision of

**Supervisor Name**

**Devyani Kamble**

****

**May 2025**

**School of Engineering**

**Ajeenkya DY Patil University, Pune**

**May 15, 2025**

**CERTIFICATE**

This is to certify that the dissertation entitled “**FitGenius**” is a Bonafide work of **Darshan Jain(2023-M-11081998) & Astha Agrawal (2023-M-30082002)** submitted to the School of Engineering, Ajeenkya D Y Patil University, Pune in partial fulfilment of the requirement for the award of the degree of **“Master of Computer Applications”**.

**Dr. Devyani Kamble**

Supervisor

**Internal-Examiner/s  External Examiner**

**Dr. Prashant Kumbharkar**

Dean, School of Computer Engineering



**April 19, 2024**

**Supervisor’s Certificate**

This is to certify that the dissertation entitled **“FitGenius”** submitted by, **Darshan Jain(2023-M-11081998) & Astha Agrawal (2023-M-30082002)** is a record of original work carried out by them under my supervision and guidance in partial fulfilment of the requirements of the degree of **Master of Computer Applications** at **School of Engineering**, **Ajeenkya DY Patil University, Pune, Maharashtra-412105**. Neither this dissertation nor any part of it has been submitted earlier for any degree or diploma to any institute or university in India or abroad.

**Dr. Devyani Kamble**

Supervisor

4

**Declaration of Originality**

We, **Darshan Jain (2023-M-11081998) & Astha Agrawal (2023-M-30082002)**, hereby declare that this dissertation entitled **“FitGenius*”*** presents our original work carried out as master's students of the School of Engineering, Ajeenkya D Y Patil University, Pune, Maharashtra. To the best of our knowledge, this dissertation contains no material previously published or written by another person, nor any material presented by us for the award of any degree or diploma of Ajeenkya D Y Patil University, Pune, or any other institution. Any contribution made to this project by others, with whom we have worked at Ajeenkya D Y Patil University, Pune, or elsewhere, is explicitly acknowledged in the dissertation. Works of other authors cited in this dissertation have been duly acknowledged under the sections “Reference” or “Bibliography”. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated, or falsified any idea/data/fact/source in our submission.

We are fully aware that in case of any non-compliance detected in the future, the Academic Council of Ajeenkya D Y Patil University, Pune, may withdraw the degree awarded to us on the basis of the present dissertation.

**Date: May 15 , 2025**

**Place:** Lohegaon, Pune

**Darshan Jain Astha Agrawal**



**Acknowledgement**

We would like to express our sincere gratitude to Devyani Kamble for her unwavering support, invaluable guidance, and profound mentorship throughout the development journey of the FitGenius project. Devyani Kamble deep expertise, keen insights, and constructive feedback have been the cornerstone of our project's success.

Her dedication to nurturing our understanding, refining our methodologies, and fostering a collaborative environment has significantly enriched our learning experience and propelled us towards achieving our goals.

Additionally, our warm thanks extend to Ajeenkya D Y Patil University, Pune, Maharashtra, for providing us with the opportunity to carry out this prestigious project and enhance our learning in various technical fields.

**Date: May 15, 2025**

**Darshan Jain (2023-M-11081998) Astha Agrawal (2023-M-30082002)**

**Place:** Lohegaon, Pune

**Index**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Contents** | **Page No.** |
| **Chapter 1** | **INTRODUCTION** | 9 |
| 1.1 Existing System | 11 |
| 1.2 Problem Definition- Need for Computerization | 13 |
| **Chapter 2** | **PROPOSED SYSTEM** | 15 |
| 2.1 Proposed System | 16 |
| 2.2 Objectives of the System | 17 |
| 2.3 Operating Environment – Hardware and Software | 19 |
| **Chapter 3** | **ANALYSIS AND DESIGN** | 24 |
| Module List | 25 |
| 3.1 Diagrams  1. Data Flow Diagram  2. Entity Relationship Diagram  3. Use Case Diagram | 30 |
| 3.2 Table Design | 38 |
| 3.3 Data Dictionary | 41 |
| 3.4 Screen Shots | 43 |
| 3.5 Report | 47 |
| 3.6 Test Procedures and Implementation | 51 |
| **Chapter 4** | **USER MANUAL** | 58 |
| 4.1 User Manual | 58 |
| 4.2 Menu Explanation | 60 |

|  |  |  |
| --- | --- | --- |
| **Chapter 5** | **CONCLUSION** | 65 |
| 5.1 Limitations & Drawbacks | 65 |
| 5.2 Future Enhancement | 67 |
| 5.3 Conclusion | 69 |
| 5.4 Code | 70 |
| 5.5 References & Bibliography | 80 |

**Chapter 1**

**INTRODUCTION**

In an era where health and fitness have become central to individual well-being, the efficient management of gym operations plays a crucial role in delivering high-quality services and personalized fitness experiences. The FitGenius Gym Management System is a comprehensive digital solution designed to streamline and automate key functions of modern gym facilities. From managing memberships and tracking health metrics to assigning trainers and generating AI-powered fitness and diet plans, FitGenius represents a transformative leap toward intelligent fitness management.

As gyms and fitness centers increasingly adopt digital practices, the need for an integrated platform that consolidates health data, communication tools, and administrative processes becomes more vital. FitGenius emerges as a powerful system that not only facilitates routine gym workflows but also leverages artificial intelligence to enhance user engagement and health outcomes.

At its core, FitGenius is built on the principles of efficiency, personalization, and innovation. It enables gym administrators, trainers, and members to interact within a centralized digital ecosystem. Users can register for personalized fitness plans, receive tailored AI-generated diet and workout schedules, track BMI, and access their weekly health reports — all through a responsive, user-friendly web interface.

Furthermore, the system introduces smart automation into gym management by:

* Allowing AI powered recommendation,
* Motivating members for higher goal,
* Enabling members to receive instant, goal-oriented plans based on their health metrics.

Powered by technologies like React.js, Node.js, MongoDB Atlas, and OpenAI's API integration, FitGenius exemplifies the synergy of front-end excellence and back-end robustness. It not only modernizes traditional gym operations but also transforms them into data-driven, AI-enhanced workflows that support holistic health goals.

Through its implementation, FitGenius aims to revolutionize how fitness centers operate by reducing administrative burden, improving member satisfaction, and promoting long-term wellness. As we delve deeper into its features, architecture, and impact, this documentation will outline how FitGenius sets a new standard for fitness technology — empowering users to take control of their health in a digitally connected world.

**1.1 Existing System**

In today’s fast-paced world, health and fitness have become a top priority for individuals across all age groups. However, the systems used by many local gyms and fitness centers are still stuck in the past. Despite the growing demand for personalized fitness and wellness experiences, most gyms continue to operate using traditional or semi-digital systems that are inefficient, disconnected, and unable to meet the dynamic needs of modern members.

**1. Manual Workflows and Paper-Based Records**

Many gyms still maintain paper files or Excel sheets to record member registrations, workout routines, payments, trainer assignments, and progress tracking. These outdated methods are prone to human errors, loss of data, duplication, and inconsistency. Members often need to re-explain their health history or fitness goals every time they interact with a trainer.

***Example:***

A member joins a gym to lose weight, but the trainer assigned to them has no idea about their past injuries or dietary preferences. The plan given is generic and not suited to their needs, resulting in frustration or even injury.

**2. Generic Fitness Plans with No Personalization**

In most existing systems, fitness plans are static and generic. Members are typically offered standard routines regardless of their individual body metrics, activity levels, or goals. This one-size-fits-all approach leads to poor outcomes, lack of motivation, and eventually high drop-out rates.

***Real-life situation:***

A working professional with limited time is given the same workout plan as a college student with high energy levels and no time constraints. The mismatch leads to disengagement.

**3. Lack of Real-Time Monitoring and Support**

Trainers often handle many clients at once, and due to the lack of a centralized system, they have no way of tracking each member’s performance or history. They either rely on memory or personal notes, which makes it difficult to offer continuous guidance or adjust plans based on progress.

***Scenario:***

A member's stamina has improved after 2 months, but their trainer continues the same beginner routine due to lack of visibility into their weekly progress.

**4. Limited Communication and Coordination**

There is often a communication gap between the gym management, trainers, and members. If a trainer is unavailable or a member wants to change their schedule or goal, there’s no structured platform to handle it efficiently. This results in confusion, mismanagement, and poor service quality.

**5. No Digital Health Profiling or Integration**

With rising awareness about fitness, members want personalized advice on diet, exercise, and health management. However, current systems don’t support capturing essential health data like BMI, weight history, medical conditions, or fitness targets. Even when collected, it's not analyzed or used effectively to generate plans or track progress.

6. No Data-Driven Insights or Reports

Traditional systems don’t offer analytics to track trends like attendance, goal achievement, or trainer effectiveness. Gym admins make decisions based on guesswork rather than data, missing opportunities to improve services or offer personalized recommendations.

In summary, the existing gym systems are not only outdated but also incapable of supporting the modern, goal-driven fitness lifestyle that members expect. As a result, gyms face challenges in retention, engagement, and reputation.

This pressing need for automation, intelligence, personalization, and centralized data management forms the foundation for the development of FitGenius—a smart Gym Management System designed to address these limitations and usher gyms into the digital age.

**1.2 Problem Definition- Need of Computerization**

In today’s fast-paced digital world, the fitness industry is evolving rapidly, with increasing expectations for personalized, data-driven health and wellness experiences. However, most local gyms and fitness centers still rely on manual methods to collect and manage member health data, leading to inefficiencies and generic services.

**Problems in the Existing Manual or Semi-Digital Systems**

* **Manual Data Collection**  
  Health details such as name, age, height, weight, and goals are often recorded on paper or basic spreadsheets, making it prone to errors, duplication, and data loss.
* **Lack of Personalization**  
  Most members receive standardized workout or diet charts with no regard for their actual body composition, activity level, or fitness goals.
* **No BMI or Health Analysis Automation**  
  Body Mass Index (BMI) calculations are often skipped or done manually, leading to inconsistent health assessments.
* **No AI Integration**  
  There is no smart system to generate customized diet and exercise plans using modern technologies like AI, making the guidance feel outdated and unscientific.
* **Poor Member Experience**  
  Without a digital system, members do not receive instant insights, structured routines, or motivating content based on their own health data.
* **Difficulty in Health Progress Tracking**  
  Trainers and admins have no centralized dashboard to track the ongoing progress or history of members, limiting effective coaching or reassessment.
* **Unsecured and Unscalable Data Handling**  
  Storing user data manually or in isolated tools compromises data security, scalability, and retrieval efficiency.

**Need for Computerization in the Fitness Domain**

To overcome the above limitations, there is a clear need for a computerized solution that combines modern web technologies with artificial intelligence. Here's what computerization enables:

* **Efficient Health Data Entry and Storage**
  + Digital health forms allow structured collection of member details.
  + Ensures validation and uniqueness (e.g., mobile/email checks).
* **Automated BMI Calculation**
  + Real-time BMI index is calculated based on height and weight.
  + Provides an immediate understanding of a user’s body status (underweight, normal, overweight, obese).
* **AI-Generated Health Plans**
  + Using Gemini AI API, generate weekly diet plans, exercise schedules, and goal summaries tailored to each member.
  + Plans are generated based on personal factors like age, gender, goal, and activity level.
* **Dynamic Data Retrieval**
  + Members can be searched using their mobile number to view their previously stored plans.
  + Saves time and provides continuity in health tracking.
* **Enhanced User Engagement**
  + A modern, interactive system makes members feel valued and motivated by showing goal-specific recommendations.
* **Admin and Trainer Efficiency**
  + Admins can focus on managing users, assigning trainers, and ensuring better service.
  + Trainers can access personalized plans and guide members accordingly.
* **Scalability and Professionalism**
  + The system is scalable and gives the gym a professional digital presence, suitable for future extensions like payment handling, performance charts, or appointment scheduling.

**Chapter 2**

**PROPOSED SYSTEM**

**2.1 Proposed System**

The proposed system, **FitGenius**, is a feature-rich, AI-powered gym management web application aimed at transforming traditional fitness workflows into a modern, personalized, and automated experience. It streamlines gym member management, automates health plan generation, and enhances user engagement through intelligent features and responsive design.

The primary goals of FitGenius include:

* **Automating Health Data Collection**  
  Replace paper forms and manual entries with an intuitive digital health form that captures essential details like name, age, weight, height, activity level, and fitness goals.
* **Centralizing Health Data Storage**  
  All member information is securely stored in a cloud-based MongoDB database, ensuring accessibility, consistency, and long-term data integrity.
* **AI-Driven Personalization**  
  Integrate Google's Gemini API to automatically generate:
  + A customized **weekly diet plan**
  + A personalized **exercise schedule**
  + A goal summary based on user data
* **Empowering Trainers**  
  Allow trainers to view assigned members and their AI-generated plans so they can provide better supervision and targeted coaching.
* **Improving Member Experience**  
  Ensure that each member receives a unique, goal-oriented health plan instantly, motivating them with structured routines and visible progress tracking.
* **Enhancing Data Accuracy and Uniqueness**  
  Prevent duplication by validating phone numbers and emails, ensuring that every user profile is distinct and reliably retrievable.

**2.2 Objectives of System**

The primary objective of the proposed FitGenius system is to modernize the gym management experience by integrating health data collection, personalized planning, and artificial intelligence into a single, user-friendly web application. The system is designed to serve gym administrators, trainers, and members by automating essential operations and enhancing the quality of fitness guidance.

Key Objectives:

* **Digitize Health Data Collection and Management**
  + Provide an online health form to collect vital user information such as age, weight, height, activity level, and fitness goals.
  + Store all user data in a centralized MongoDB database for easy retrieval, analysis, and long-term record-keeping.
* **Automate Personalized Plan Generation**
  + Integrate AI (Google Gemini API) to generate customized diet plans, workout schedules, and goal summaries for each member based on their health metrics.
  + Eliminate the need for manual plan preparation by trainers or nutritionists, saving time and effort.
* **Improve Member Onboarding Experience**
  + Offer a seamless digital onboarding process where members can either retrieve their existing data or register as a new user with an intuitive interface.
  + Display AI-generated plans immediately upon registration to help users take quick action toward their fitness goals.
* **Enhance Trainer and Admin Efficiency**
  + Enable administrators to assign trainers and monitor overall member activity through backend management tools.
  + Provide trainers with access to assigned member profiles, including their generated plans, to offer informed coaching and support.
* **Ensure Data Accuracy and Uniqueness**
  + Validate user identity through unique mobile numbers and optional email addresses to avoid duplicate entries.
  + Maintain a consistent format and integrity of stored data for reliable analytics and personalized services.
* **Support Informed Decision-Making**
  + Present structured weekly plans to members, promoting routine adherence and goal tracking.
  + Help trainers and admins make better decisions by analyzing user health data and generated recommendations.
* **Enable Scalable and Secure Deployment**
  + Host the backend on Railway and the frontend on Vercel for reliable, scalable, and secure cloud-based access.
  + Ensure responsive design and optimal performance across all modern devices and browsers.
* **Foster a Personalized Fitness Culture**
  + Deliver an engaging and individualized experience to each gym member, increasing motivation and long-term participation.
  + Promote healthy lifestyle changes through actionable, AI-powered recommendations tailored to each user’s specific goals and physical profile.

**2.3 User Requirements**

The successful implementation of the **FitGenius** platform hinges on a clear understanding of its diverse user base, ranging from casual visitors to dedicated gym members. By addressing their individual needs and expectations, FitGenius seeks to deliver a modern, engaging, and data-driven fitness experience. Below is a detailed breakdown of the key user requirements for the system:

**1. General Visitors (Website Users)**

* **Ease of Access:** The platform is designed to be publicly accessible without login, offering open access to all its core features including BMI calculation, pricing plans, and workout insights.
* **Responsive Design:** Users require an intuitive and mobile-friendly interface that works seamlessly across devices and screen sizes.
* **Motivational UI/UX:** The website must be visually appealing and include inspirational content and motivational prompts to encourage users to explore services and begin their fitness journey.
* **Fast Navigation:** Smooth and quick access to core sections like “Start Your Journey” and “Discover Your Plan” from the landing page ensures a positive first impression and higher engagement.

**2. New Gym Members**

* **Health Data Input:** Members require a simple and secure interface to input their personal health data, including weight, height, age, gender, activity level, and fitness goals.
* **Instant Results:** After data submission, users expect immediate generation of personalized diet and workout plans powered by AI, with minimal delay or technical complexity.
* **No Registration Barrier:** Users are not required to sign up or create an account, making the onboarding process seamless and convenient.

**3. Returning Users**

* **Data Retrieval by Mobile Number:** Users who have previously submitted their health data expect to retrieve their personalized plan using only their mobile number.
* **Plan Persistence:** The system should retain and display their original AI-generated weekly fitness and diet schedule.

**4. Platform Owner / Administrator (Internal Stakeholders)**

* **Visit Tracking:** The platform includes a visitor tracking feature to log and analyze the number of daily site visits, helping assess platform engagement and popularity.
* **Advertisement Integration:** The ability to monitor traffic allows the owner to consider monetization strategies through ad placements and sponsorships.
* **Minimal Manual Maintenance:** The system is largely self-operational with minimal backend involvement, relying on automation and API integrations (like Gemini AI) to drive functionality.

**5. AI Functionality Expectations**

* **Contextual AI Responses:** The AI system must generate diet and workout recommendations tailored to the specific inputs provided by users.
* **Clarity and Structure:** Responses from the AI should be well-formatted, easy to understand, and structured into separate sections for diet, workout, and goal overview.
* **Consistency and Accuracy:** Repeated inputs with the same parameters should yield consistent outputs to maintain trust in AI reliability.

**2.4 Operating Environment – Hardware and Software**

FitGenius is developed as a cloud-hosted web application with minimal dependence on local hardware infrastructure. However, the following hardware specifications are necessary for development, hosting, and usage:

* **Processor:** Minimum Intel Core i3 or equivalent AMD processor; Core i5 or higher recommended for development environments.
* **RAM:** Minimum 4 GB RAM for general use; 8 GB or more recommended for local development, especially when running Node.js and database servers concurrently.
* **Storage:** Minimum 100 GB of disk space, especially for development and logging purposes. Live deployments typically use cloud-based storage solutions.
* **Network**: Stable internet connectivity is essential for accessing the hosted system and for integrating AI APIs like Gemini.
* **Display:** A monitor with a minimum resolution of 1280×800 for development and responsive design testing.
* **Input Devices:** Standard keyboard and mouse are sufficient.
* **Backup :** Cloud-based or external storage is recommended for code and database backup during development or testing phases.

**Software Requirements**

FitGenius uses modern web technologies, cloud platforms, and AI integration to deliver a dynamic and intelligent fitness experience. The software stack includes:

* **Operating System:** Cross-platform support for Windows, Linux, and macOS for development.
* **Frontend Framework:** React.js with Vite as the build tool, offering fast refresh and optimized production builds.
* **Backend Environment:** Node.js with Express.js as the web server framework.
* **Database**: MongoDB (Atlas cloud-hosted) for storing user health data, AI-generated plans, and visit tracking.
* **AI Integration:** Gemini AI API (via Google Generative Language API) for generating personalized health and fitness plans.
* **Package Manager:** npm (Node Package Manager) for managing dependencies.
* **API Communication:** Axios for HTTP requests between frontend and backend.
* **Environment Variables:** .env file used for secure configuration of API keys, MongoDB URI, and server ports.
* **Email Service:** Nodemailer integrated with Gmail SMTP for sending contact form messages or future OTP functionality.
* **Version Control:** Git with GitHub integration for version control, team collaboration, and CI/CD deployment.
* **Web Browsers:** Chrome, Firefox, Edge, Safari — responsive across modern browsers.
* Hosting Platforms:
  + **Frontend:** Vercel (React frontend with custom environment variables and build optimization).
  + **Backend:** Railway (Node.js Express backend connected to MongoDB Atlas).
* **Monitoring and Logs:** Railway’s built-in monitoring tools or third-party logging like LogRocket or Sentry (optional).

**Cloud and Network Considerations**

* **API Hosting and Access:** Must support HTTPS endpoints for secure AI API access and RESTful interactions.
* **Environment Scalability:** Both frontend (Vercel) and backend (Railway) are deployed on scalable platforms that auto-handle traffic loads.

Chapter 3

ANALYSIS AND DESIGN

The design and development of FitGenius are based on the core principles of accessibility, automation, intelligence, and user-centricity. Unlike traditional gym management systems that often involve restricted admin access or manual processes, FitGenius takes a modern approach by embracing open accessibility and leveraging AI to deliver personalized experiences.

A. System Analysis

FitGenius is designed to solve several modern challenges faced by fitness enthusiasts and gym-goers in a digital-first era. Here's a breakdown of the core analysis points:

* Open Access Model:
  + Unlike admin-controlled portals, FitGenius is open to the public.
  + Users can directly interact with the platform without needing prior approval or login for basic services.
  + This eliminates barriers to entry and encourages exploration and user engagement.
* AI-Powered Personalization:
  + By collecting health metrics like age, weight, height, and goals, the system generates instant, customized fitness and diet plans using the Gemini AI API.
  + Users do not need human trainers or consultations to get started, saving time and effort.
* **User Behavior Tracking:**
  + The system records visits to the site using backend tracking mechanisms.
  + This visit data helps in analyzing user interest and can be used for:
    - Targeted content.
    - Personalized offers.
    - Advertising and monetization strategies.
* **Visual Design Focus:**
  + A highly attractive, responsive, and interactive frontend has been developed using React.
  + Smooth animations, engaging UI, and mobile-first responsiveness make the website modern and appealing.
  + This design helps retain users and encourages longer interaction time, which improves conversion rates.
* **Business and Monetization Possibilities:**
  + FitGenius not only helps users with health goals but also serves as a lead magnet for future business plans.
  + Through visit tracking and high-quality UX, the platform creates an opportunity for placing third-party fitness ads or partnerships, generating passive revenue.

**B. Design Considerations**

1. **Frontend Design:**
   * Built with React.js for high performance and component-based scalability.
   * Styled using CSS with animations, ensuring aesthetic and intuitive interfaces.
   * Hero section includes impactful call-to-action buttons ("Start Your Journey", "Discover Your Plan") that smoothly scroll users to key sections like Pricing and Workout.
2. **Backend Architecture:**
   * Developed using Node.js and Express for lightweight and fast server response.
   * API-based architecture for modular data access.
   * MongoDB Atlas is used as a NoSQL database to store health forms, AI-generated plans, and visit logs.
3. **AI Integration:**
   * Google Gemini API is integrated on the backend to generate weekly diet and workout plans.
   * Inputs collected from users are passed to the Gemini prompt, and the structured response is stored and displayed.
4. **User Flow Design:**
   * User visits → Enters phone number → Searches or creates profile → AI processes data → Plan displayed.
   * This seamless and minimal-click flow enhances usability and reduces user dropout.
5. **Visit Recording Logic:**
   * A visit counter API is triggered on every load of the site.
   * This information is stored in the database for admin-level analysis (e.g., how many people visited today).
6. **Security and Maintenance:**
   * All sensitive data such as API keys and database URIs are securely managed through .env files.
   * Email functionality uses secure SMTP configurations through nodemailer.
   * All user input is validated before processing.

**3.1 Diagrams**

**1. Data Flow Diagram (DFD)**

A Data Flow Diagram (DFD) is a graphical representation of the flow of data through an information system, depicting its process aspects. It serves as a preliminary step to provide an overview of the system's functionality, which can later be elaborated upon. DFDs are commonly utilized for visualizing data processing in structured design methodologies.

DFDs illustrate the types of data that enter and exit the system, as well as their origins and destinations. They depict how data moves through various processes and where it is stored within the system. However, DFDs do not convey information regarding the timing of processes or whether they operate sequentially or concurrently.

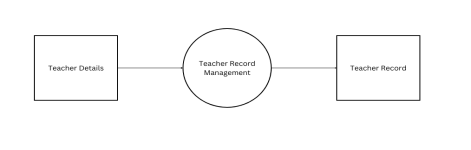
In essence, a DFD serves as a visual tool to understand the flow of data within a system, aiding in the analysis, design, and communication of system requirements. It helps stakeholders comprehend the data transformation processes and identify potential areas for optimization or enhancement within the system architecture.

**Level 0:**

• This represents the highest level of abstraction in the system, often referred to as the context level.

• At this level, you depict the entire system as a single process or entity, without delving into its internal processes.

• In your case, "Teacher Record Management" would be the main process or system being depicted at this level.

**Fig: 1 DFD Level- 0**

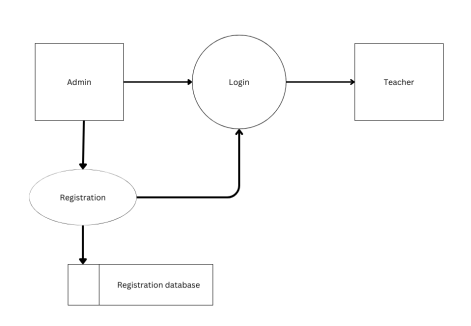
30

**Level 1:**

• Level 1 DFDs break down the main process/system into its major subprocesses or subsystems.

• Each subprocess is represented as a separate process box connected to the main process/system.

• For example, "Login," "Admin," "Teacher," "Registration," etc., would be the major subprocesses under "Teacher Record Management" at Level 1.

**Fig :2 DFD Level -1**

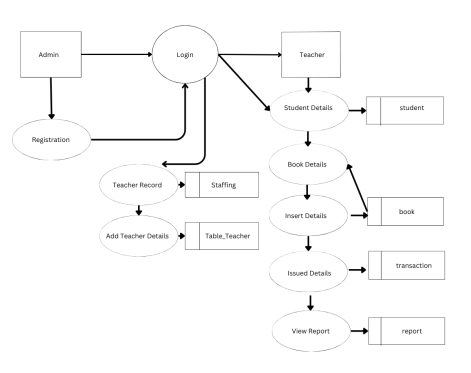
31

**Level 2:**

• Level 2 DFDs further decompose the major subprocesses identified at Level 1 into more detailed processes.

• Each subprocess from Level 1 is expanded to show its internal processes or data flows. • For instance, under "Admin" subprocess, you may have processes like "Staffing," "Table\_Teacher," "Student," "Transaction," "Book," "Report," etc.

• Similarly, other subprocesses such as "Teacher" and "Registration" would be further detailed at this level.

**Fig: 3 DFD Level -2**

32

**2. Entity Relationship Diagram (ERD)**

An Entity Relationship Diagram (ERD) is a visual representation of the entities (or objects), attributes, and relationships within a system. It helps to illustrate the structure of a database or information system by showing how data is organized and how entities relate to each other. In the following diagram, we have entities represented as rectangles, with attributes listed within each entity. Relationships between entities are depicted using lines connecting them, with optional symbols indicating cardinality and participation constraints.

The ERD provided implements:

**Entities:**

1. Admin

2. Teacher

3. User

4. Subject

**Attributes for each entity:**

• For Admin: ID, RegDate, MobileNumber, Email, AdminName, UserName, Password • For Teacher: ID, JoiningDate, MobileNumber, Email, Qualification, TeacherSub, Address • For User: ID, Name, Picture, MobileNumber, Email

• For Subject: ID, Name, CreationDate

**Relationships between entities:**

**1. Admin Has TRMS:**

• Indicates a one-to-one relationship between Admin and TRMS entities. • Each Admin manages exactly one instance of TRMS, and each instance of TRMS is managed by exactly one Admin.

**2. Admin Adds Teacher:**

• Indicates a one-to-many relationship between Admin and Teacher entities. • Each Admin can add multiple teachers to the system, but each teacher is added by only one Admin.

**3. Teacher Belongs to Subject:**

• Indicates a many-to-many relationship between Teacher and Subject entities.

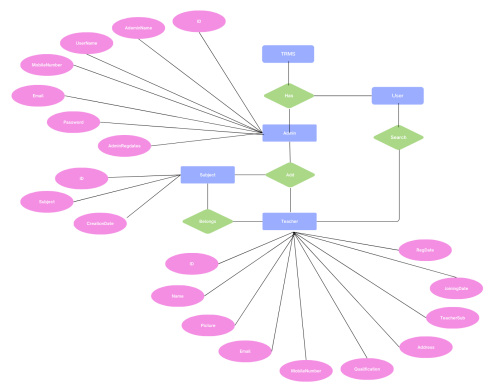
33

• Each Teacher can belong to multiple subjects, and each subject can have multiple teachers teaching it.

**4. User Searches Subject:**

• Indicates a many-to-many relationship between User and Subject entities. • Each User can search for multiple subjects, and each subject can be searched by multiple users.

**ER-Diagram:**

**Fig : 4 ER-Diagram**

34

**3. Use Case Diagram**

A Use Case Diagram is a graphical representation of the interactions between actors (users or external systems) and the system being designed. It illustrates the various use cases or functionalities of the system and how different actors interact with those use cases. In the following Use Case Diagram, we have actors represented as stick figures, use cases represented as ovals, and relationships between actors and use cases depicted using lines. The provided Use Case Diagram implements:

**Actors:**

• Admin

• User

**Use Cases:**

1. Add Teacher

2. Dashboard

3. Add Subject

4. Manage Subject (Update Details)

5. Manage Teacher (Update Details)

6. Search (Teacher)

7. Generate Reports

8. Update Profile

9. Change Password

10. Password Recovery

**Explanation of Use Cases:**

**1. Add Teacher:**

• This use case represents the functionality for adding a new teacher to the system. • Admin actor initiates this use case to add teacher details to the system.

35

**2. Dashboard:**

• Represents the main dashboard interface of the system.

• Both Admin and User actors interact with the dashboard to access various functionalities.

**3. Add Subject:**

• Represents the functionality for adding a new subject to the system. • Admin actor initiates this use case to add subject details to the system. **4. Manage Subject (Update Details):**

• Represents the functionality for managing subject details, including updating existing subject information.

• Admin actor initiates this use case to update subject details in the system. **5. Manage Teacher (Update Details):**

• Represents the functionality for managing teacher details, including updating existing teacher information.

• Admin actor initiates this use case to update teacher details in the system. **6. Search (Teacher):**

• Represents the functionality for searching for teachers in the system. • Both Admin and User actors can initiate this use case to search for teachers based on specific criteria.

**7. Generate Reports:**

• Represents the functionality for generating various reports from the system data. • Admin actor initiates this use case to generate reports for analysis or decision making purposes.

**8. Update Profile:**

• Represents the functionality for updating user profiles.

• Both Admin and User actors can initiate this use case to update their profile information.

**9. Change Password:**

• Represents the functionality for changing user passwords.

• Both Admin and User actors can initiate this use case to change their login passwords.

36

**10. Password Recovery:**

• Represents the functionality for recovering forgotten passwords.

• Both Admin and User actors can initiate this use case to recover their forgotten passwords through a recovery process.

**Use Case Diagram**

**Fig: 5 Use Case Diagram**

37

**3.2 Table Design**

A table is a fundamental data structure used to organize information into rows and columns. It serves as a foundation for storing and presenting data in a structured format. For instance, databases utilize tables to efficiently store and retrieve data, enabling quick access to specific information from designated rows. Similarly, websites employ tables to present data in a structured manner, facilitating the display of multiple rows of information on a single page. Spreadsheets also leverage tables to combine the functions of storing and displaying data in a structured format.

In database design, multiple tables are typically employed, each serving a distinct purpose. For instance, in a company database, separate tables may be designated for employees, clients, and suppliers, with each table tailored to accommodate specific data requirements. Within these tables, each field corresponds to a column, representing a particular data attribute, while each entry (or record) corresponds to a row, encapsulating a collection of related data.

To access specific data from a table, queries are formulated to retrieve information from individual columns and rows based on predefined criteria. This structured approach to data organization facilitates efficient data management and retrieval within database systems. Table design is a fundamental aspect of database development, crucial for organizing and storing data efficiently. It involves structuring data into tables composed of rows and columns, each representing a specific entity or attribute. The primary goal of table design is to create a logical and coherent schema that accurately reflects the relationships between different data elements.

When designing tables, it's essential to identify and define the various entities and attributes that need to be stored in the database. This involves analyzing the requirements of the system and determining the appropriate data types, constraints, and relationships for each table. For example, in a Teachers Record Management System, tables may include entities such as teachers, subjects, and administrative staff, with attributes such as names, IDs, and contact information.

Normalization is another key aspect of table design, aimed at reducing data redundancy and improving data integrity. By organizing data into separate tables and establishing relationships

38

between them, normalization ensures that each piece of information is stored in only one place, minimizing the risk of inconsistencies or errors.

In addition to normalization, table design should also consider factors such as data integrity constraints, indexing strategies, and performance optimization techniques. Data integrity constraints, such as primary keys, foreign keys, and unique constraints, help maintain the accuracy and consistency of the data. Indexing can improve query performance by enabling faster data retrieval, while optimization techniques such as denormalization may be used to enhance performance in certain scenarios.

Furthermore, table design should adhere to best practices and standards to ensure compatibility, scalability, and maintainability. This includes following naming conventions for tables and columns, documenting the schema comprehensively, and considering future expansion or modification requirements.

Overall, effective table design is essential for creating a robust and efficient database schema that meets the needs of the Teachers Record Management System. By carefully planning and implementing the structure of tables, you can lay a solid foundation for storing, retrieving, and managing data effectively within the system.

Moreover, table design should also take into account the security requirements of the system. This involves implementing access controls and permissions to restrict unauthorized access to sensitive data stored in the tables.

**1. Table Name: Admin**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.no** | **Field name** | **Type** | **Null** | **Constraint** |
| 1. | id | Int(20) | No | Not Null |
| 2. | Uname | Varchar(20) | No | Not Null |
| 3. | psw | Varchar(20) | No | Not Null |

39

**2. Table Name : Staffing**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Constraint** |
| id | Int(20) | No | PRIMARY KEY |
| Name | Varchar(30) | No | Null |
| Gender | Varchar(20) | No | Null |
| Dob | Varchar(20) | No | Null |
| Email | Varchar(40) | No | Null |
| Mob | Varchar(10) | No | Null |
| Mst | Varchar(10) | No | Null |
| File | Varchar(20) | No | Null |
| Address | Varchar(40) | No | Null |
| Nat | Varchar(30) | No | Null |
| Anum | Varchar(50) | No | Null |
| Uname | Varchar(20) | No | Null |

**3.Table Name : table\_teacher**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Constraint** |
| Name | Varchar(20) | No | Null |
| Id | Int(20) | No | PRIMARY KEY |
| Email | Varchar(40) | No | Null |
| Qualification | Varchar(20) | No | Null |
| Addewss | Varchar(50) | No | Null |
| Teacher Sub | Varchar(20) | No | Null |
| description | Varchar(30) | No | Null |

40

**3.3 Data Dictionary**

A Data Dictionary is a crucial component of any system, providing a comprehensive reference guide for the data elements used within the system. In the context of a Teacher Record Management System (TRMS), the Data Dictionary outlines the key data elements along with their descriptions, types, and other relevant attributes.

Following is the Data Dictionary tailored for a TRMS:

|  |  |  |
| --- | --- | --- |
| **FIELD NAME** | **DATA TYPE** | **DESCRIPTION** |
| Id | Int | Teacher id |
| Email | Varchar | Email id of the teacher |
| Password | Varchar | Password for admin login, teacher login |
| Gender | Varchar | Admin , teacher gender |
| Dob | Date | Admin , teacher dob |
| Picture | Varchar | Image of the teacher |
| Mob | Int | Mobile number of the teacher |
| Mst | Varchar | Married status of the teacher |
| Address | Varchar | Teacher address |
| Nat | Varchar | Teacher native |
| Anum | Int | Income of the teacher |
| Uname | Varchar | User-name of the teacher |

A Data Dictionary serves as a vital resource in system development, providing a concise reference for the various data elements utilized within the Teacher Record Management System (TRMS). It offers a structured overview of key attributes such as teacher ID, email, password, gender, date of birth, picture, mobile number, marital status, address, native details, income, and username. Each element is meticulously described, specifying its data type and significance within the system's functionality.

41

A Data Dictionary serves as an indispensable tool in system development, providing a structured repository of essential data elements crucial for the functioning of the Teacher Record Management System (TRMS). Within this comprehensive guide, each data attribute is meticulously documented, including its name, data type, and detailed description.

In the TRMS context, the Data Dictionary encapsulates vital information such as teacher IDs, email addresses, passwords, gender, dates of birth, images, contact numbers, marital statuses, addresses, native details, incomes, and usernames. These elements form the backbone of the system, facilitating the effective management and organization of teacher records.

By offering a standardized framework for understanding and referencing data elements, the Data Dictionary streamlines communication and collaboration among system developers, ensuring consistency and coherence across all stages of development. Moreover, it serves as a reference point for system maintenance and updates, enabling developers to make informed decisions regarding data structures and modifications.

The clarity and specificity provided by the Data Dictionary enhance system transparency and facilitate ease of use for administrators, teachers, and other stakeholders involved in TRMS operations. Additionally, it promotes accuracy and reliability by minimizing ambiguity and discrepancies in data interpretation.

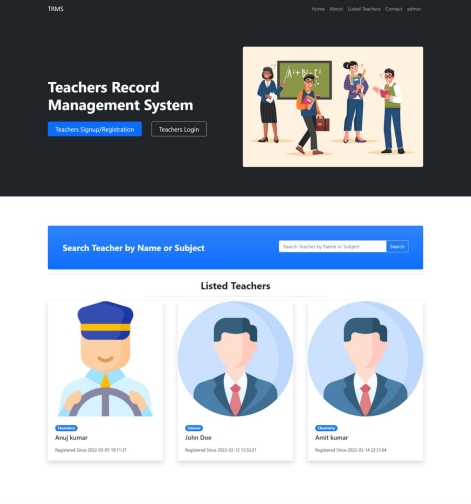
As an essential component of the TRMS development process, the Data Dictionary contributes to the system's overall robustness and resilience, enabling it to effectively meet the evolving needs and requirements of educational institutions. Its role extends beyond initial development, serving as a valuable resource for ongoing system enhancements and optimization efforts.

Furthermore, the Data Dictionary serves as a cornerstone for ensuring compliance with data governance standards and regulatory requirements. By providing a comprehensive reference for data elements and their attributes, it facilitates adherence to data privacy laws and regulations governing the handling and storage of sensitive information within the TRMS.

42

**3.4 Screenshots**

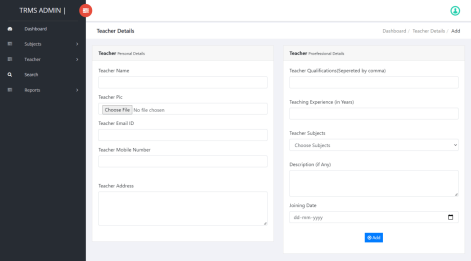
**Home-Page:**

****43

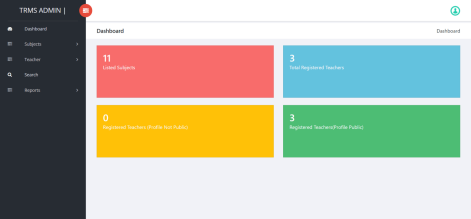
**Teacher Signup**

****

**Add Teachers-Page**

****44

**Admin Dashboard**

**Admin sign-in page**

****45

**About-us page**

****

**Contact page**

****46

**3.5 Reports**

The Reports section of the Teacher Record Management System (TRMS) is a vital component that enables administrators and educators to generate and access various types of reports for informed decision-making and assessment purposes. This section encompasses a range of reporting functionalities designed to provide valuable insights into teacher-related data, academic performance, and administrative processes.

Within the Reports section, users can generate diverse reports, including but not limited to:

1. **Teacher Performance Reports:** These reports offer an overview of teacher performance metrics, such as attendance records, teaching evaluations, and professional development activities. Administrators can assess individual teacher performance and identify areas for improvement or recognition.

2. **Subject-wise Analysis:** This type of report allows users to analyze teaching assignments and performance across different subjects or disciplines. It provides valuable insights into subject-specific trends, student engagement levels, and academic outcomes.

3. **Enrolment and Demographic Reports:** These reports provide data on student enrolment trends, demographic distributions, and class sizes. They assist administrators in resource allocation, class scheduling, and strategic planning initiatives.

4. **Financial Reports:** Financial reports within the TRMS offer insights into budget allocations, expenditure patterns, and resource utilization related to teacher salaries, professional development expenses, and classroom resources. These reports aid in financial planning and budget management.

5. **Compliance and Regulatory Reports:** Compliance reports help educational institutions ensure adherence to regulatory requirements, accreditation standards, and data privacy laws. They provide evidence of compliance with policies and regulations governing teacher certification, background checks, and curriculum standards.

47

The Reports section of the TRMS empowers users with customizable reporting options, allowing them to specify parameters, filters, and timeframes to generate tailored reports based on their specific needs and objectives. By leveraging data analytics and visualization tools, the TRMS facilitates data-driven decision-making, performance monitoring, and accountability across all levels of the educational institution.

Overall, the Reports section serves as a valuable tool for administrators, educators, and stakeholders, enabling them to access actionable insights, track progress, and drive continuous improvement within the educational ecosystem.

The Reports section of the Teacher Record Management System (TRMS) is more than just a repository of data; it's a dynamic tool that empowers administrators, educators, and stakeholders to make informed decisions and foster continuous improvement within the educational ecosystem.

With customizable reporting options and a diverse range of report types available, the Reports section enables users to access actionable insights tailored to their specific needs and objectives. Whether it's assessing teacher performance, analyzing enrollment trends, or ensuring regulatory compliance, users can leverage these reports to gain a comprehensive understanding of various aspects of the educational institution.

By tracking key metrics and performance indicators, stakeholders can monitor progress towards organizational goals and identify areas for enhancement. This data-driven approach not only enhances transparency and accountability but also facilitates evidence-based decision-making at all levels of the institution.

Moreover, the Reports section serves as a catalyst for driving continuous improvement initiatives within the educational ecosystem. By identifying trends, patterns, and areas of concern, stakeholders can implement targeted interventions and strategic interventions to address challenges and capitalize on opportunities for growth and development.

Furthermore, the accessibility and user-friendliness of the Reports section ensure that stakeholders across the organization can easily access and interpret relevant data. This

48

democratization of data fosters collaboration, communication, and collective problem-solving, ultimately leading to a more cohesive and effective educational environment.

Reports in the context of a Teacher Record Management System (TRMS) play a pivotal role in providing valuable insights, tracking progress, and facilitating decision-making for administrators, educators, and other stakeholders. These reports serve as essential tools for analyzing data, identifying trends, and assessing the effectiveness of various initiatives within the educational ecosystem. One of the primary functions of reports is to provide administrators with a comprehensive overview of teacher-related metrics and performance indicators.

These may include metrics such as the total number of teachers, subjects taught, teacher qualifications, experience levels, and professional development activities. By consolidating this information into concise and informative reports, administrators gain a deeper understanding of the teacher landscape within their institution.

Reports also enable administrators to track the progress of key initiatives and programs aimed at improving teaching and learning outcomes. For example, administrators may generate reports to monitor the implementation of professional development programs, assess their impact on teacher performance, and identify areas for improvement. These reports serve as valuable tools for evaluating the effectiveness of educational interventions and refining strategies to better support teacher development.

In addition to tracking teacher-related metrics, reports in a TRMS also provide insights into student outcomes and academic performance. By analyzing data on student achievement, attendance, and behavior, administrators can identify patterns, trends, and areas of concern that may require intervention. These insights inform decision-making processes aimed at enhancing student success and addressing educational disparities.

Reports generated by a TRMS are not only valuable for internal stakeholders but also serve as communication tools for external audiences such as parents, regulatory bodies, and funding agencies. Administrators may use these reports to demonstrate accountability, transparency, and compliance with educational standards and regulations.

49

Furthermore, reports generated by a TRMS facilitate data-driven decision-making at all levels of the educational institution. By presenting data in a clear, visual format, these reports empower administrators, educators, and policymakers to make informed decisions about resource allocation, curriculum development, and strategic planning. Whether identifying areas for improvement, allocating budgetary resources, or assessing program effectiveness, TRMS reports serve as essential tools for evidence-based decision-making.

To ensure the effectiveness and relevance of reports generated by a TRMS, it is essential to consider the specific informational needs of different stakeholders. Reports should be tailored to address the unique requirements of administrators, educators, students, parents, and other stakeholders, providing them with actionable insights that support their respective roles and responsibilities within the educational ecosystem.

In summary, reports generated by a TRMS play a multifaceted role in supporting administrative decision-making, enhancing teacher effectiveness, improving student outcomes, and fostering collaboration within the educational community. By harnessing the power of data analytics and visualization, TRMS reports empower stakeholders to drive continuous improvement and innovation in teaching and learning practices.

TRMS reports also contribute to the overall efficiency and effectiveness of administrative processes within educational institutions. By automating the generation and distribution of reports, TRMS streamlines administrative workflows, reduces manual data entry errors, and enhances data accuracy and reliability. This automation frees up valuable time and resources that administrators can allocate to more strategic tasks, such as analyzing report findings, identifying trends, and developing targeted interventions to address areas of improvement.

Moreover, TRMS reports serve as essential tools for monitoring compliance with regulatory requirements and accreditation standards. Educational institutions must adhere to various regulations and standards set forth by government agencies, accreditation bodies, and other stakeholders

50

**3.6 Test Procedures and Implementation**

Testing is a group of techniques aimed at determining the correctness of an application according to predefined scripts. However, it's important to note that testing alone cannot uncover all defects within an application. Its primary objective is to identify failures so that they can be rectified. Testing does not guarantee flawless functionality under all conditions; rather, it reveals areas where the application may not perform as expected under specific conditions. Testing encompasses both code examination and execution in diverse environments, considering various aspects of the code.

In modern software development practices, testing is often carried out separately from the development process. This separation allows the testing team to provide valuable insights that can inform and improve the software development process. Testing is a systematic evaluation process aimed at determining whether a system or its components meet specified requirements. Essentially, it involves executing a system to uncover any discrepancies, errors, or deviations from the intended requirements.

The segregation of testing from the development process in modern software development practices not only facilitates a dedicated focus on quality assurance but also enables a more comprehensive evaluation of the system's functionality and performance. By conducting testing as a separate activity, specialized testing teams can meticulously assess the software against predefined criteria and scenarios, ensuring thorough validation and verification. This structured approach to testing enhances the reliability, robustness, and overall quality of the software, ultimately contributing to the delivery of a superior product that aligns closely with user expectations and business objectives.

In modern software development, testing is often conducted independently of the development process. This separation allows specialized testing teams to thoroughly assess the software against predefined criteria and scenarios.

51

**Types of Testing :**

• **Unit Testing**: Focuses on testing individual units or components of the software in isolation.

• **Integration Testing**: Evaluates the interaction and integration between different units or components to ensure they function correctly together.

• **Acceptance Testing**: Validates whether the software meets the specified business requirements and is ready for deployment.

**Explanation :**

1. **Unit Testing**:

• **Definition**: Unit testing is a testing technique that focuses on testing individual units or components of the software in isolation. A unit can be a function, method, or class. • **Purpose**: The primary goal of unit testing is to validate that each unit of the software performs as expected according to its design and requirements. It helps identify bugs or defects within individual units before they are integrated into larger parts of the system.

• **Process**: During unit testing, developers write test cases to verify the behavior of each unit. These test cases typically cover various scenarios, including typical inputs, boundary conditions, and error conditions. Automated testing frameworks are often used to automate the execution of these test cases.

• **Benefits**: Unit testing facilitates early bug detection, simplifies debugging and troubleshooting, and promotes code maintainability and modularity. It also provides developers with confidence in the reliability and correctness of their code. 2. **Integration Testing**:

• **Definition**: Integration testing is a testing technique that evaluates the interaction and integration between different units or components of the software. It verifies that these units work together as expected when integrated into larger parts of the system.

• **Purpose**: The main objective of integration testing is to uncover defects or inconsistencies that may arise when integrating multiple units or components. It

52

ensures that the integrated system behaves as intended and that data flows smoothly between its various components.

• **Process**: Integration testing can be conducted using various approaches, such as top down, bottom-up, or a combination of both. Test cases are designed to simulate interactions between different units or components, covering both normal and exceptional scenarios.

• **Benefits**: Integration testing helps detect interface issues, data flow errors, and compatibility issues early in the development process. It promotes system reliability, stability, and interoperability by validating the integration of different software modules.

3. **Acceptance Testing**:

• **Definition**: Acceptance testing is a testing technique that verifies whether the software meets the specified business requirements and is ready for deployment. It involves testing the system from an end-user perspective to ensure it fulfills its intended purpose.

• **Purpose**: The primary goal of acceptance testing is to validate that the software satisfies the expectations and needs of its stakeholders, including customers, users, and business owners. It ensures that the software delivers the desired functionality and meets quality standards before it is released to production.

• **Process**: Acceptance testing is typically conducted by end-users or stakeholders in a real or simulated environment. Test cases are designed based on user stories, requirements, or acceptance criteria defined during the requirements gathering phase. The focus is on verifying that the software meets user needs, is user-friendly, and delivers the expected business value.

• **Benefits**: Acceptance testing provides assurance that the software meets business objectives and user expectations. It helps identify any discrepancies between the software and user requirements early in the development lifecycle, reducing the risk of costly rework or dissatisfaction among stakeholders.

53

**Validation**

Validation is a critical aspect of testing, especially in web development projects like the Teachers Record Management System. Here are some validation scenarios specific to this project: 1. **Form Field Validation**: Ensure that all form fields, such as those used for user registration, teacher information, and queries, are validated to accept only appropriate inputs. For example:

• Check for required fields and display error messages if they are left blank. • Validate email addresses to ensure they follow the correct format.

• Validate phone numbers to accept only numeric characters and specific formats. • Validate date fields to ensure they are entered in a valid date format.

2. **Data Integrity Validation**: Validate the integrity of data stored in the database. For example:

• Ensure that unique constraints are enforced for fields like email addresses and usernames to prevent duplicate entries.

• Validate foreign key constraints to ensure that references to related tables are valid.

3. **Authentication and Authorization Validation**: Validate the authentication and authorization mechanisms implemented in the system. For example:

• Test login functionality with valid and invalid credentials.

• Test access controls to ensure that users can only access features and data appropriate to their role (e.g., administrators accessing admin-only functions). 4. **Search Functionality Validation**: Validate the search functionality to ensure it returns accurate results. For example:

• Test searching for teachers by name or subject to verify that relevant results are returned.

• Test edge cases, such as searching for non-existent or partially entered names, to ensure the system handles them gracefully.

5. **Report Generation Validation**: Validate the report generation functionality to ensure it produces accurate reports based on specified criteria. For example:

• Test generating reports for different time periods to verify that the correct number of teachers added during that period is displayed.

54

• Test generating reports with various filters and parameters to ensure they produce the expected results.

6. **Error Handling Validation**: Validate the error handling mechanisms implemented in the system. For example:

• Test submitting forms with invalid data to ensure that appropriate error messages are displayed.

• Test edge cases and boundary conditions to ensure that the system gracefully handles unexpected inputs and errors.

7. **Cross-Browser and Cross-Device Validation**: Validate the system's compatibility with different web browsers and devices. For example:

• Test the system on popular web browsers (e.g., Chrome, Firefox, Safari) to ensure consistent behavior and appearance.

• Test the system on different devices (e.g., desktop, tablet, mobile) to ensure responsiveness and usability across various screen sizes.

By thoroughly testing these validation scenarios, you can ensure that the Teachers Record Management System functions reliably, accurately, and securely, meeting the needs and expectations of its users.

Additionally, it's crucial to conduct boundary testing to validate the system's behavior at the extremes of its input ranges. This involves testing with minimum and maximum valid inputs as well as inputs just beyond these limits to ensure the system behaves as expected. Furthermore, stress testing can be performed to assess the system's performance under heavy load or high traffic conditions. By systematically validating these aspects of the Teachers Record Management System, you can enhance its robustness, reliability, and user satisfaction.

Test procedures and implementation are crucial components of the software development lifecycle, ensuring that the developed system meets its specified requirements and functions reliably in diverse operating environments. This phase involves planning, executing, and documenting various tests to validate the functionality, performance, and usability of the software.

55

The first step in test procedures and implementation is to define the testing strategy and establish test objectives. This includes identifying the types of tests to be conducted, such as unit testing, integration testing, system testing, and acceptance testing. Additionally, test procedures should outline the criteria for success and specify the resources, tools, and timelines required for testing activities.

Once the testing strategy is established, the next step is to develop test cases and test scenarios based on the system requirements and user stories. Test cases outline the specific inputs, expected outputs, and steps to be executed during testing, while test scenarios describe the sequence of actions and interactions that simulate real-world usage scenarios. These artifacts serve as a roadmap for executing tests and evaluating the system's behavior under different conditions.

With test cases and scenarios in place, the testing team proceeds to execute the tests according to the predefined procedures. This involves running the software, inputting test data, observing the system's responses, and comparing the actual outcomes against expected results. Testing may be conducted manually or automated, depending on the complexity of the system and the availability of testing tools.

Throughout the testing process, it is essential to track and record test results systematically. This includes documenting any issues, defects, or discrepancies identified during testing and prioritizing them based on severity and impact. Test reports and logs provide valuable feedback to developers, enabling them to address issues promptly and iteratively improve the software.

In addition to functional testing, test procedures should also encompass non-functional aspects such as performance, security, and usability. Performance testing evaluates the system's responsiveness, scalability, and resource utilization under varying loads, while security testing assesses vulnerabilities and safeguards sensitive data. Usability testing focuses on the user experience, ensuring that the system is intuitive, accessible, and meets the needs of its intended users.

Finally, test procedures and implementation involve conducting acceptance testing with end-users or stakeholders to validate that the software meets their expectations and requirements. User

56

acceptance testing (UAT) allows stakeholders to interact with the system in a real-world environment and provide feedback on its functionality, usability, and suitability for their needs. In summary, effective test procedures and implementation are essential for ensuring the quality, reliability, and performance of software systems. By following structured testing processes, organizations can identify and address defects early in the development lifecycle, mitigate risks, and deliver high-quality software that meets user expectations.

Test Procedures and Implementation Overview :

1. Test procedures should be comprehensive, covering all functional and non-functional aspects of the software system, including functionality, performance, security, usability, and reliability.

2. Implementation of test procedures should follow a systematic approach, starting from test planning and design, through test execution, to test reporting and analysis. 3. Test cases and scenarios should be designed to validate each requirement and use case, ensuring that the software meets its intended objectives and user expectations. 4. Test procedures should incorporate both manual and automated testing techniques to maximize test coverage and efficiency while minimizing human error.

5. Continuous feedback and iteration are essential for refining test procedures and improving the overall quality of the software through iterative testing cycles.

57

**Chapter 4**

**USER MANUAL**

**4.1 User Manual**

The User Manual for the Teacher Record Management System (TRMS) provides comprehensive guidance on navigating and utilizing the system's features effectively.

1. **Dashboard**:

• **Overview**: The Dashboard serves as the central hub for users, providing an overview of essential information such as upcoming events, recent notifications, and quick access to key features.

• **Usage**: Upon logging in, users are greeted with the Dashboard, where they can quickly access various modules and functionalities of the TRMS.

2. **Subjects**:

• **Overview**: The Subjects section allows users to manage subjects within the system, including adding new subjects, updating existing ones, or removing outdated records. • **Usage**: Users can easily add new subjects to the system, update subject details, or delete obsolete subjects as needed, ensuring accurate subject records.

3. **Teachers**:

• **Overview**: The Teachers section empowers users to manage teacher records, including adding new teachers, updating their information, or removing outdated records.

• **Usage**: Users can seamlessly add new teachers to the system, update their personal and professional details, or deactivate records for retired or non-active teachers. 4. **Search**:

• **Overview**: The Search feature enables users to search for teachers by entering their names or specific criteria, facilitating quick and efficient retrieval of teacher records. • **Usage**: Users can utilize the Search functionality to find specific teacher records based on various search criteria, streamlining data retrieval tasks.

58

5. **Reports**:

• **Overview**: The Reports section allows users to generate and view reports, providing insights into various aspects such as teacher demographics, subject distribution, or recruitment trends.

• **Usage**: Users can generate reports to analyze trends, track progress, and make informed decisions based on data collected within the TRMS.

6. **Profile**:

• **Overview**: The Profile section enables users to manage their personal profile information within the TRMS, including updating contact details, uploading profile pictures, or modifying login credentials.

• **Usage**: Users can maintain their profile information up-to-date, ensuring accurate user profiles and enhancing system security.

7. **Change Password**:

• **Overview**: The Change Password feature allows users to update their login passwords securely.

• **Usage**: Users can change their passwords regularly to enhance security and protect their accounts from unauthorized access.

8. **Logout**:

• **Overview**: The Logout button enables users to securely log out of the TRMS, terminating their session and preventing unauthorized access to their accounts. • **Usage**: Users can click the Logout button to end their session and safely exit the TRMS when their tasks are complete.

9. **Forgot Password**:

• **Overview**: The Forgot Password feature provides users with a mechanism to reset their passwords in case they forget them.

• **Usage**: Users can initiate the password reset process by providing their registered email ID and following the instructions provided to regain access to their accounts.

The Teacher Record Management System (TRMS) is a comprehensive platform designed to streamline administrative processes and enhance efficiency in educational institutions. By

59

centralizing teacher-related data and providing intuitive user interfaces, TRMS aims to simplify teacher management tasks and improve overall productivity.

TRMS offers various modules tailored to the needs of different user roles within the educational ecosystem. The Dashboard serves as a centralized hub for accessing key information and functionalities, such as monitoring subject and teacher statistics, accessing reports, and managing user profiles.

With the Subjects module, administrators can manage subject records, including adding new subjects, updating details, and ensuring accurate subject information within the system. Similarly, the Teachers module empowers administrators to oversee teacher records, from adding new teachers to updating their personal and professional details.

The Search functionality enhances user experience by enabling quick and efficient retrieval of teacher records based on specific criteria. Additionally, the Reports module provides valuable insights into teacher demographics, subject distribution, and recruitment trends, aiding administrators in strategic decision-making processes.

User profiles can be managed through the Profile section, allowing users to update their personal information, upload profile pictures, and modify login credentials. The Change Password feature ensures security by enabling users to update their passwords regularly.

Ultimately, TRMS facilitates seamless communication and collaboration among stakeholders, fosters data-driven decision-making, and supports the broader goals of educational institutions in achieving excellence in teaching and learning. Through its array of features and functionalities, TRMS creates a cohesive environment where educators, administrators, students, and parents can interact, share information, and work towards common goals.

**4.2 Menu Explanation**

In the context of the Teacher Record Management System (TRMS), the menu serves as a navigational interface through which users interact with the system's functionalities. Each menu

60

option corresponds to a specific feature or module within the TRMS, tailored to the roles and responsibilities of different user types, such as administrators, teachers, and users.

The "Dashboard" option provides users with a centralized view of essential information, such as the total number of subjects and teachers registered in the system. This feature offers admins and teachers alike a quick snapshot of key metrics, facilitating informed decision-making and efficient management of resources.

"Subjects" and "Teachers" options empower admins to manage subject and teacher records, respectively. Admins can add new subjects, update existing ones, add new teachers, and manage teacher details through these functionalities. This ensures the accuracy and relevance of data within the system, contributing to effective teacher record management.

The "Search" option enables admins to swiftly locate specific teacher records by entering their names. This search functionality streamlines the process of retrieving relevant information, saving time and enhancing user experience within the TRMS.

"Report" functionality equips admins with valuable insights into teacher recruitment trends and other relevant data. By generating reports on various aspects of teacher records, admins can analyze patterns, identify areas for improvement, and make data-driven decisions to optimize teacher management processes.

Overall, the menu options in the TRMS provide users with intuitive access to essential features and functionalities, promoting seamless navigation and efficient management of teacher records. Whether it's adding new subjects, managing teacher profiles, or generating insightful reports, the menu empowers users to accomplish their tasks effectively within the TRMS ecosystem.

**Admin Module:**

1. **Dashboard**:

• Provides a comprehensive overview for admins, displaying total subjects and teachers. 2. **Subjects**:

• Allows admin to manage subjects by adding or updating subject details.

61

3. **Teachers**:

• Enables admin to add new teachers and manage existing teacher profiles. 4. **Search**:

• Allows admin to search for teachers by name, facilitating quick access to specific records.

5. **Report**:

• Provides admin with reports on teacher recruitment trends and other relevant data. 6. **Profile**:

• Enables admin to update their own profile information as needed.

7. **Change Password**:

• Allows admin to change their login password for enhanced security.

8. **Logout**:

• Logs admin out of the system securely.

9. **Forgot Password**:

• Allows admin to reset their password using registered email ID and contact number.

The Admin Module within the Teacher Record Management System (TRMS) is a pivotal component designed to empower administrators with the tools and functionalities necessary for efficient management of teacher records and related administrative tasks. At its core, the Admin Module serves as a centralized hub where administrators can oversee and administer various aspects of the system.

**Teacher Module:**

1. **Dashboard**:

• Serves as a welcome page for teachers, providing key information at a glance. 2. **Queries**:

• Enables teachers to view queries raised by users and respond as necessary. 3. **Profile**:

• Allows teachers to update their personal information within the system. 4. **Change Password**:

• Enables teachers to change their login password securely.

62

5. **Logout**:

• Logs teachers out of the system securely.

The Teacher Module within the Teacher Record Management System (TRMS) is designed to provide educators with a user-friendly interface to manage their profiles, address queries, and interact with the system's functionalities seamlessly. Tailored to meet the specific needs of teachers, this module offers a range of features to facilitate efficient communication and self

management.

**Users:**

1. **Search Teachers**:

• Users can search for teachers by entering subject names, facilitating easy access to relevant teacher profiles.

2. **Raise Queries**:

• Users can raise queries, which are visible to teachers for response and resolution.

Through its array of features and functionalities, TRMS creates a cohesive environment where educators, administrators, students, and parents can interact, share information, and work towards common goals. Educators can leverage TRMS to collaborate on lesson planning, share instructional materials, and exchange best practices, thereby enhancing the quality of teaching and learning experiences.

A user manual is paramount for the successful utilization of software systems, serving as a comprehensive guide to assist users in navigating the intricacies of its functionalities, features, and operations. It acts as a vital link between software developers and end-users, facilitating effective communication and knowledge transfer.

At its core, the manual provides an introductory overview of the software system, offering users insights into its purpose, key features, and intended audience. This initial section establishes a solid foundation, enabling users to grasp the context and relevance of the manual in relation to their specific usage needs and objectives.

Moving forward, the manual dives into meticulous explanations of various modules, functionalities, and workflows within the software. Each section meticulously presents step-by-

63

step instructions, often accompanied by relevant screenshots or illustrations, to guide users through specific tasks or operations, ensuring clarity and ease of understanding.

Furthermore, the manual incorporates a comprehensive troubleshooting guide and frequently asked questions (FAQs) section, aiming to address common issues or concerns users may encounter while navigating the software. This proactive approach empowers users to troubleshoot and resolve issues independently, thereby enhancing their overall experience and satisfaction.

In addition to procedural instructions, the manual may integrate best practices, tips, and recommendations aimed at optimizing user productivity and efficiency. These insights provide users with valuable guidance on leveraging advanced features or functionalities of the software effectively, ultimately maximizing their utility and value.

Moreover, the manual underscores the significance of clear organization and intuitive navigation, ensuring users can easily locate the information they require. This user-centric design enhances usability and user satisfaction, thereby contributing to a positive overall experience with the software.Continuous updates and revisions to the manual are imperative to align with software updates, feature enhancements, and user feedback. By maintaining an up-to-date manual, software developers can ensure users have access to accurate and relevant information, fostering continued engagement and success with the software.

64

**Chapter 5**

**CONCLUSION**

**5.1 Limitations & Drawbacks**

To begin our exploration of the Teacher Record Management System (TRMS), it's essential to understand the pivotal role that efficient teacher record management plays in the educational landscape. In today's digital age, educational institutions are increasingly relying on technology driven solutions to streamline administrative processes, enhance collaboration, and optimize resource allocation. TRMS emerges as a comprehensive web-based platform designed to address the diverse needs of educational administrators, teachers, and stakeholders alike. This system aims to revolutionize the way teacher records are managed, providing a centralized and integrated platform for storing, accessing, and analysing vital information related to educators. By harnessing the power of technology, TRMS endeavours to empower educational institutions to achieve their goals of academic excellence, student success, and operational efficiency. In this journey of discovery, let us delve deeper into the intricacies of TRMS, uncovering its potential to reshape the educational landscape and empower educators and administrators to thrive in the digital era.

To which the Limitations & Drawbacks are :

1. **Dependence on Technology:** The Teacher Record Management System (TRMS) relies heavily on technology infrastructure, including servers, databases, and network connectivity. Any disruptions or failures in these technological components can significantly impact the system's availability and performance.

2. **Security Vulnerabilities:** Despite implementing security measures such as encryption and access controls, TRMS may still be vulnerable to security breaches, including unauthorized access, data leaks, and cyberattacks. These vulnerabilities pose a risk to the confidentiality, integrity, and availability of sensitive teacher and user data.

3. **User Adoption Challenges:** Introducing a new system like TRMS may face resistance from users who are accustomed to traditional methods of record-keeping and communication. Encouraging user adoption and overcoming resistance to change can be a significant challenge, requiring effective communication, training, and support initiatives.

65

4. **Data Accuracy and Integrity:** Maintaining accurate and up-to-date data within TRMS requires diligent data entry and management practices. Human errors, inconsistencies, and outdated information can compromise the reliability and usefulness of the system, leading to erroneous decisions and inefficient operations.

5. **Scalability Concerns:** As the number of teachers, subjects, and users within the educational institution grows, TRMS must be able to scale seamlessly to accommodate increased data volumes and user interactions. Ensuring scalability requires careful planning and ongoing infrastructure investments to support the system's expanding needs.

6. **Integration Complexity:** Integrating TRMS with existing systems, such as student information systems or learning management systems, can be complex and time consuming. Incompatibilities, data migration challenges, and customization requirements may arise during the integration process, delaying deployment and increasing implementation costs.

7. **Maintenance Overhead:** Sustaining TRMS over time entails ongoing maintenance activities, including software updates, bug fixes, and performance optimizations. The maintenance overhead can strain resources and require dedicated personnel and budget allocations to ensure the system remains operational and up-to-date.

8. **Regulatory Compliance:** Educational institutions must adhere to various regulations and standards related to data privacy, security, and accessibility. Ensuring TRMS compliance with these regulations, such as GDPR or FERPA, requires ongoing monitoring, documentation, and audit procedures, adding complexity to system management.

9. **User Support Demands:** Providing adequate support services to users of TRMS, including administrators, teachers, and students, is essential for ensuring smooth system operation and user satisfaction. Addressing user inquiries, resolving technical issues, and delivering training sessions may strain support resources and require efficient support workflows.

10. **Cost Considerations:** Implementing and maintaining TRMS involves significant financial investments, including software licensing fees, infrastructure expenses, personnel costs, and ongoing operational expenditures. Balancing the costs with the expected benefits and return on investment (ROI) requires careful financial planning and risk assessment.

66

**5.2 Future Enhancements**

As we envision the future trajectory of the Teacher Record Management System (TRMS), several potential enhancements emerge to further augment its capabilities and address evolving needs in the educational landscape:

1. **Advanced Analytics and Insights:** Enhancing TRMS with advanced analytics capabilities can enable administrators to derive actionable insights from teacher data. By implementing data visualization tools, predictive analytics, and trend analysis features, TRMS can empower administrators to make data-driven decisions, identify areas for improvement, and optimize resource allocation effectively.

2. **Integration with Learning Management Systems (LMS):** Integrating TRMS with existing Learning Management Systems can facilitate seamless data exchange between teacher records and student learning data. This integration can enable educators to better track student progress, tailor teaching strategies, and provide personalized support, fostering a more holistic approach to education.

3. **Mobile Accessibility:** Developing mobile applications or responsive web interfaces for TRMS can extend its accessibility beyond desktop environments. Mobile access enables administrators and teachers to manage records, access reports, and communicate efficiently while on the go, enhancing productivity and flexibility in educational workflows.

4. **Enhanced Collaboration Features:** Introducing collaborative features within TRMS, such as discussion forums, shared document repositories, and real-time messaging capabilities, can foster collaboration and communication among administrators, teachers, and other stakeholders. These features facilitate knowledge sharing, professional development, and community building within the educational ecosystem.

5. **Integration with Human Resource Systems:** Integrating TRMS with Human Resource Management Systems (HRMS) can streamline processes related to teacher recruitment, onboarding, and performance evaluation. By syncing data between TRMS and HRMS, educational institutions can ensure seamless management of teacher profiles, qualifications, and employment records.

6. **Automated Workflows and Notifications:** Implementing automated workflows and notifications within TRMS can streamline administrative tasks and enhance efficiency.

67

Automated processes for tasks such as teacher onboarding, performance appraisals, and document approvals can reduce manual workload and ensure timely completion of tasks. 7. **Enhanced Security Measures:** Continuously enhancing security measures within TRMS to safeguard sensitive teacher and student data against cyber threats and unauthorized access. Implementing robust encryption protocols, multi-factor authentication, and regular security audits can strengthen TRMS's resilience against security breaches. 8. **Customization and Scalability:** Providing customizable features and scalability options within TRMS to accommodate the unique needs and growth trajectories of different educational institutions. Modular architecture, customizable user interfaces, and flexible data structures enable TRMS to adapt to evolving requirements and scale seamlessly as educational institutions expand.

By embracing these future enhancements, TRMS can evolve into a dynamic and indispensable tool for educational institutions worldwide, empowering them to navigate the complexities of modern education effectively and achieve their overarching goals of academic excellence and student success.

In addition to enhancing operational efficiency and facilitating data-driven decision-making, the implementation of these future enhancements underscores TRMS's commitment to fostering innovation and adaptability in the educational sector. By staying at the forefront of technological advancements and responding to the evolving needs of educators and administrators, TRMS solidifies its position as a catalyst for positive change in education.

Furthermore, the continuous development and refinement of TRMS reflect a dedication to excellence and a proactive approach to addressing emerging challenges in education. As educational institutions strive to remain competitive and meet the demands of a rapidly changing world, TRMS stands ready to support their journey towards continuous improvement, empowering educators, administrators, and students to thrive in the digital age and beyond.

68

**5.3 Conclusion**

In conclusion, the Teacher Record Management System (TRMS) represents a transformative solution poised to revolutionize teacher record management practices in educational institutions worldwide. Through its comprehensive features, user-friendly interface, and commitment to innovation, TRMS empowers administrators, educators, and stakeholders to streamline administrative processes, enhance collaboration, and optimize resource allocation.

By leveraging advanced technologies such as data analytics, mobile accessibility, and integration with existing systems, TRMS provides a robust platform for managing teacher records efficiently and effectively. Its ability to adapt to the evolving needs of educational institutions and support their goals of academic excellence and student success positions TRMS as a vital tool in modern education.

As we look towards the future, the continued development and enhancement of TRMS promise to further elevate its impact and relevance in the educational landscape. By embracing innovation, collaboration, and a relentless pursuit of excellence, TRMS sets a new standard for teacher record management, empowering educational institutions to thrive in an ever-changing world.

In essence, TRMS represents not just a software solution, but a catalyst for positive change in education, enabling educators and administrators to unlock new efficiencies, drive continuous improvement, and ultimately, shape a brighter future for students around the globe.

Moreover, TRMS embodies a commitment to inclusivity, accessibility, and equity in education, ensuring that every student receives the support and resources they need to succeed. By facilitating data-driven decision-making and personalized learning experiences, TRMS paves the way for a more equitable and student-centric educational ecosystem.

In the journey towards educational excellence, TRMS stands as a beacon of innovation and progress, guiding educators and administrators towards a future where every student has the opportunity to reach their full potential.

69

**5.4 Code**

**Sample Code:**

<?php

session\_start();

error\_reporting(0);

include('includes/dbconnection.php');

if (strlen($\_SESSION['trmsaid']==0)) {

header('location:logout.php');

} else {

?>

<!doctype html>

<html class="no-js" lang="en">

<head>

<title>TRMS Admin Dashboard</title>

<link rel="apple-touch-icon" href="apple-icon.png">

<link rel="stylesheet" href="../vendors/bootstrap/dist/css/bootstrap.min.css"> <link rel="stylesheet" href="../vendors/font-awesome/css/font-awesome.min.css"> <link rel="stylesheet" href="../vendors/themify-icons/css/themify-icons.css"> <link rel="stylesheet" href="../vendors/flag-icon-css/css/flag-icon.min.css"> <link rel="stylesheet" href="../vendors/selectFX/css/cs-skin-elastic.css"> <link rel="stylesheet" href="../vendors/jqvmap/dist/jqvmap.min.css">

<link rel="stylesheet" href="../assets/css/style.css">

70

<link href='https://fonts.googleapis.com/css?family=Open+Sans:400,600,700,800' rel='stylesheet' type='text/css'>

</head>

<body>

<?php include\_once('includes/sidebar.php');?>

<div id="right-panel" class="right-panel">

<?php include\_once('includes/header.php');?>

<!-- Header-->

<div class="breadcrumbs">

<div class="col-sm-4">

<div class="page-header float-left">

<div class="page-title">

<h1>Dashboard</h1>

</div>

</div>

</div>

<div class="col-sm-8">

<div class="page-header float-right">

<div class="page-title">

<ol class="breadcrumb text-right">

<li class="active">Dashboard</li>

</ol>

</div>

71

</div>

</div>

</div>

<div class="content mt-3">

<a href="manage-subjects.php">

<div class="col-sm-6 col-lg-6">

<div class="card text-white bg-flat-color-4">

<div class="card-body pb-0">

<div class="dropdown float-right">

</div>

<?php

$sql ="SELECT ID from tblsubjects ";

$query = $dbh -> prepare($sql);

$query->execute();

$results=$query->fetchAll(PDO::FETCH\_OBJ);

$sublist=$query->rowCount();

?>

<h2 class="mb-0">

<span class="count"><?php echo htmlentities($sublist);?></span> </h2>

<p class="text-light">Listed Subjects</p>

<div class="chart-wrapper px-3" style="height:70px;" height="70"> <canvas id="widgetChart4"></canvas>

</div>

72

</div>

</div>

</div>

</a>

<!--/.col-->

<a href="manage-teacher.php">

<div class="col-sm-6 col-lg-6">

<div class="card text-white bg-flat-color-2">

<div class="card-body pb-0">

<div class="dropdown float-right">

</div>

<?php

$sql1 ="SELECT ID from tblteacher ";

$query1 = $dbh -> prepare($sql1);

$query1->execute();

$results1=$query1->fetchAll(PDO::FETCH\_OBJ);

$totalteacher=$query1->rowCount();

?>

<h2 class="mb-0">

<span class="count"><?php echo htmlentities($totalteacher);?></span> </h2>

<p class="text-light">Total Registered Teachers</p>

<div class="chart-wrapper px-0" style="height:70px;" height="70"> <canvas id="widgetChart2"></canvas>

</div>

</div>

</div>

73

</div>

</a>

<a href="manage-notpublicprofileteacher.php">

<div class="col-sm-6 col-lg-6">

<div class="card text-white bg-flat-color-3">

<div class="card-body pb-0">

<div class="dropdown float-right">

</div>

<?php

$sql1 ="SELECT ID from tblteacher where isPublic is null || isPublic='0'"; $query1 = $dbh -> prepare($sql1);

$query1->execute();

$results1=$query1->fetchAll(PDO::FETCH\_OBJ);

$totalteacher=$query1->rowCount();

?>

<h2 class="mb-0">

<span class="count"><?php echo htmlentities($totalteacher);?></span> </h2>

<p class="text-light">Registered Teachers (Profile Not Public)</p>

<div class="chart-wrapper px-0" style="height:70px;" height="70"> <canvas id="widgetChart2"></canvas>

</div>

</div>

</div>

</div>

</a>

74

<a href="manage-publicprofileteacher.php">

<div class="col-sm-6 col-lg-6">

<div class="card text-white bg-flat-color-5">

<div class="card-body pb-0">

<div class="dropdown float-right">

</div>

<?php

$sql1 ="SELECT ID from tblteacher where isPublic='1'";

$query1 = $dbh -> prepare($sql1);

$query1->execute();

$results1=$query1->fetchAll(PDO::FETCH\_OBJ);

$totalteacher=$query1->rowCount();

?>

<h2 class="mb-0">

<span class="count"><?php echo htmlentities($totalteacher);?></span> </h2>

<p class="text-light">Registered Teachers(Profile Public)</p>

<div class="chart-wrapper px-0" style="height:70px;" height="70"> <canvas id="widgetChart2"></canvas>

</div>

</div>

</div>

</div>

</a>

</div> <!-- .content -->

</div><!-- /#right-panel -->

75

<!-- Right Panel -->

<script src="../vendors/jquery/dist/jquery.min.js"></script>

<script src="../vendors/popper.js/dist/umd/popper.min.js"></script> <script src="../vendors/bootstrap/dist/js/bootstrap.min.js"></script> <script src="../assets/js/main.js"></script>

<script src="../vendors/chart.js/dist/Chart.bundle.min.js"></script> <script src="../assets/js/dashboard.js"></script>

<script src="../assets/js/widgets.js"></script>

<script src="../vendors/jqvmap/dist/jquery.vmap.min.js"></script> <script src="../vendors/jqvmap/examples/js/jquery.vmap.sampledata.js"></script> <script src="../vendors/jqvmap/dist/maps/jquery.vmap.world.js"></script> <script>

(function($) {

"use strict";

jQuery('#vmap').vectorMap({

map: 'world\_en',

backgroundColor: null,

color: '#ffffff',

hoverOpacity: 0.7,

selectedColor: '#1de9b6',

enableZoom: true,

showTooltip: true,

values: sample\_data,

scaleColors: ['#1de9b6', '#03a9f5'],

normalizeFunction: 'polynomial'

});

})(jQuery);

</script>

76

</body>

</html>

<?php } ?>

**Purpose of code :**

The dashboard file (**dashboard.php**) serves as the main page for the TRMS Admin Dashboard. It provides an interface to view various statistics and information related to subjects and teachers registered in the system.

**Dependencies**

• **dbconnection.php**: This file contains the database connection details and functions required for database interaction.

• External CSS and JavaScript libraries:

• Bootstrap

• Font Awesome

• Themify Icons

• Flag Icon CSS

• SelectFX CSS

• jqvmap CSS

• Internal CSS and JavaScript files:

• **style.css**: Custom CSS styles for the dashboard.

• **dashboard.js**: JavaScript file for dashboard functionalities.

• **widgets.js**: JavaScript file for dashboard widgets.

**PHP Code Overview**

• **Session Management**:

• The **session\_start()** function initializes a new session or resumes the existing one. • Error reporting is suppressed by setting **error\_reporting(0)**.

77

• **Database Connection**:

• The **dbconnection.php** file is included to establish a connection to the database. • **HTML Structure**:

• The HTML structure of the dashboard page is defined within the **<body>** section. **Dashboard Statistics**

• **Listed Subjects**:

• Retrieves the total number of subjects listed in the system from the database. • Displays the count of listed subjects along with a descriptive text.

• **Total Registered Teachers**:

• Retrieves the total number of registered teachers from the database.

• Displays the count of registered teachers along with a descriptive text.

• **Registered Teachers (Profile Not Public)**:

• Retrieves the total number of registered teachers whose profiles are not public from the database.

• Displays the count of such teachers along with a descriptive text.

• **Registered Teachers (Profile Public)**:

• Retrieves the total number of registered teachers whose profiles are public from the database.

• Displays the count of such teachers along with a descriptive text.

**JavaScript Functionality**

• The JavaScript code initializes a vector map (**#vmap**) with specified configurations and sample data for visualization.

**Script and Library Inclusions**

• Various JavaScript libraries such as jQuery, Bootstrap, and Chart.js are included to provide additional functionalities and visualizations.

• Custom JavaScript files (**dashboard.js** and **widgets.js**) are included for specific dashboard functionalities.

**Note:** The PHP code at the end of the file checks if the user is logged in. If not, it redirects the user to the logout page.

78

**Output**

The output of the PHP code in the **dashboard.php** file is an HTML page that serves as the main page for the TRMS Admin Dashboard. Here's a summary of the expected output: 1. **HTML Structure**:

• The HTML structure includes elements such as **<head>**, **<title>**, **<body>**, and various **<div>** containers for organizing content.

2. **CSS and JavaScript Dependencies**:

• External CSS and JavaScript files, such as Bootstrap, Font Awesome, and custom stylesheets and scripts, are included to style and enhance the functionality of the dashboard.

3. **Sidebar and Header**:

• The dashboard includes a sidebar and a header, which are likely included from separate files (**includes/sidebar.php** and **includes/header.php**, respectively). 4. **Breadcrumb Navigation**:

• Breadcrumb navigation is displayed at the top of the page, indicating the current location within the dashboard.

5. **Dashboard Statistics**:

• The dashboard displays various statistics related to subjects and teachers in card format. Each card contains information such as the total number of listed subjects, total registered teachers, registered teachers with public profiles, and registered teachers with non-public profiles. These statistics are retrieved from the database using SQL queries embedded within the PHP code.

6. **Vector Map**:

• If a vector map (**#vmap**) is present on the page, it is initialized with specific configurations and sample data for visualization. However, the output may vary depending on the availability of the vector map and the specific data used. 7. **JavaScript Functionality**:

• JavaScript functionalities provided by included libraries and custom scripts are applied to enhance user interaction and visualizations on the dashboard.

Overall, the output is a visually appealing and informative dashboard interface for administrators to monitor and manage subjects and teachers in the TRMS system.

79

**5.5 References & Bibliography**

**References:**

1. Tutorialspoint. “PHP Tutorial”.

URL: https://www.tutorialspoint.com/php/index.htm

2. W3Schools. “PHP Tutorial”.

URL: https://www.w3schools.com/php/

3. PHP.net. “PHP Manual”.

URL: https://www.php.net/manual/en/index.php

4. Codecademy. “Learn PHP”

URL: https://www.codecademy.com/learn/learn-php

5. MySQL Tutorial. "MySQL Tutorial”

URL: https://www.mysqltutorial.org/

6. Stack Overflow. “PHP tagged questions”.

URL:https://stackoverflow.com/questions/tagged/php

**Bibliography:**

1. Modern PHP: New Features and Good Practices by Josh Lockhart. 2. Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5 by Robin Nixon. 3. PHP and MySQL for Dynamic Web Sites: Visual QuickPro Guide by Larry Ullman. 4. PHP Objects, Patterns, and Practice by Matt Zandstra.

5. MySQL Explained: Your Step-by-Step Guide by Andrew Comeau and Apress.

80