**NurtureNest**

*Mini Project Report*

*Submitted by*

**Ashly Mol Thomas**

**Reg. No.: AJC00MCA-I020**

*In Partial fulfillment for the Award of the Degree of*

**INTEGRATED MASTER OF COMPUTER APPLICATIONS**

**(INMCA)**

**APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY**



**AMAL JYOTHI COLLEGE OF ENGINEERING**

**KANJIRAPPALLY**

[Affiliated to APJ Abdul Kalam Technological University, Kerala. Approved by AICTE, Accredited by NAAC. Koovappally, Kanjirappally, Kottayam, Kerala – 686518]

# 2024-2025

## DEPARTMENT OF COMPUTER APPLICATIONS

### AMAL JYOTHI COLLEGE OF ENGINEERING

**KANJIRAPPALLY**



**CERTIFICATE**

This is to certify that the Project report, “**NURTURENEST”** is the bona fide work of **ASHLY MOL THOMAS (Regno: AJC00MCA-I020)** in partial fulfillment of the requirements for the award of the Degree of Integrated Master of Computer Applications under APJ Abdul Kalam Technological University during the year 2023-24.

**Ms. Merin Manoj Mr. Binumon Joseph**

**Assistant professor Assistant Professor**

**Internal Guide Coordinator**

**Rev. Fr. Dr. Rubin Thottupurathu Jose**

**Head of the Department**

**DECLARATION**

I hereby declare that the project report **“NURTURENEST”** is a bona fide work done at Amal Jyothi College of Engineering, towards the partial fulfilment of the requirements for the award of the **Integrated Master of Computer Applications (MCA)** from **APJ Abdul Kalam Technological University**, during the academic year **2024-2025**.

**Date: ASHLY MOL THOMAS**

**KANJIRAPPALLY Reg: AJC00MCA-0000**

# ACKNOWLEDGEMENT

First and foremost, I thank God almighty for his eternal love and protection throughout the project. I take this opportunity to express my gratitude to all who helped me in completing this project successfully. It has been said that gratitude is the memory of the heart. I wish to express my sincere gratitude to our Director (Administration) **Rev. Fr. Dr. Roy Abraham Pazhayaparampil** and Principal **Dr. Lillykutty Jacob** for providing good faculty for guidance.

I owe a great depth of gratitude towards our Head of the Department **Rev.Fr.Dr. Rubin Thottupurathu Jose** for helping us. I extend my whole hearted thanks to the project coordinator **Mr. Binumon Joseph** for his valuable suggestions and for overwhelming concern and guidance from the beginning to the end of the project. I would also express sincere gratitude to my guide **Ms. Merin Manoj** for her inspiration and helping hand.

I thank our beloved teachers for their cooperation and suggestions that helped me throughout the project. I express my thanks to all my friends and classmates for their interest, dedication, and encouragement shown towards the project. I convey my hearty thanks to my family for the moral support, suggestions, and encouragement to make this venture a success.

ASHLY MOL THOMAS

# ABSTRACT

.

NurtureNest is a robust, web-based platform designed to streamline the management of baby health and developmental milestones for parents and healthcare providers. It enables parents to easily schedule their children's vaccination appointments, receive timely immunization reminders, and access complete vaccination records. The platform goes beyond physical health tracking by offering feeding suggestions tailored to a child’s growth phase, as well as mental health insights based on their developmental stage. NurtureNest incorporates machine learning to optimize healthcare provider workflows. Upon uploading a vaccine image, the system automatically extracts and displays all necessary information, ensuring accurate documentation. By leveraging cutting-edge technology, NurtureNest empowers parents and medical professionals to manage every aspect of a child’s well-being effectively, providing personalized updates and recommendations.

Languages: PYTHON DJANGO, HTML, AJAX, JAVASCRIPT, CSS, JQUERY

Database: SQLite

1. User Module (Parents)

NurtureNest provides parents with a complete suite of tools to manage their child's health:

* Registration and Login: Parents can securely sign up and access the platform.
* Profile Completion: Parents can maintain comprehensive profiles for their children, including personal information and medical history.
* Vaccine Scheduling: Parents can schedule vaccinations for their children and receive reminders via email or SMS for upcoming appointments.
* Notifications: Parents are alerted about upcoming vaccinations, health check-ups, and essential updates.
* Feeding Chart: Provides feeding suggestions tailored to the child’s age, offering personalized guidance on nutrition.
* Mental Health Tips: Parents receive mental health tips based on their child’s developmental stage, supporting emotional and cognitive growth.

2. Admin Module

The Admin Module offers a set of functionalities to ensure the platform runs efficiently and securely:

* Vaccine Management: Admins can add, update, and manage the vaccination schedule, ensuring all vaccines are available for parents and healthcare providers.
* Feeding Chart Management: Admins can upload and update feeding recommendations to match the needs of children at different stages of growth.
* Mental Health Tips: Admins curate and manage mental health tips based on developmental phases to support parents in fostering their child’s emotional health.
* Stock Management: Admins oversee the vaccine inventory, ensuring sufficient supply and timely restocking.
* Health Center Approval: Admins are responsible for approving health centers after reviewing their applications, granting them access to the platform’s features.
* Vaccine Request Management: Admins manage and approve vaccine requests made by health centers, ensuring the availability of essential vaccines for patients.
* Patient Management: Admins have oversight over the platform's user profiles, ensuring accuracy and completeness of health records.

3. Healthcare Provider Module (Health Centers)

Healthcare providers have access to the tools necessary to deliver efficient care to their patients:

* Vaccine Requests: Providers can request vaccines from the admin, ensuring they have the necessary stock for appointments.
* Vaccine Appointment Approvals: Providers can approve or reschedule vaccination appointments requested by parents.
* Available Vaccines: Providers can view the list of available vaccines, ensuring they can meet patient needs promptly.
* Stock Management: Providers manage their vaccine stock to ensure they maintain a balanced inventory.

Machine Learning Integration: Providers can upload images of vaccines, and the platform uses machine learning to automatically extract and display all relevant details for proper documentation.

**CONTENT**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SL. NO** | | **TOPIC** | **PAGE NO** | |
| **1** | | **INTRODUCTION** |  | |
| **1.1** | | **PROJECT OVERVIEW** |  | |
| **1.2** | | **PROJECT SPECIFICATION** |  | |
| **2** | | **SYSTEM STUDY** |  | |
| **2.1** | | **INTRODUCTION** |  | |
| **2.2** | | **LITERATURE REVIEW** |  | |
| **2.3** | | **PROPOSED SYSTEM** |  | |
| **2.4** | | **ADVANTAGES OF PROPOSED SYSTEM** |  | |
| **3** | | **REQUIREMENT ANALYSIS** |  | |
| **3.1** | | **FEASIBILITY STUDY** |  | |
| **3.1.1** | | **ECONOMICAL FEASIBILITY** |  | |
| **3.1.2** | | **TECHNICAL FEASIBILITY** |  | |
| **3.1.3** | | **BEHAVIORAL FEASIBILITY** |  | |
| **3.1.4** | | **FEASIBILITY STUDY QUESTIONNAIRE** |  | |
| **3.2** | | **SYSTEM SPECIFICATION** |  | |
| **3.2.1** | | **HARDWARE SPECIFICATION** |  | |
| **3.2.2** | | **SOFTWARE SPECIFICATION** |  | |
| **3.3** | | **SOFTWARE DESCRIPTION** |  | |
| **3.3.1** | | **PYTHON DJANGO** |  | |
| **3.3.2** | | **SQLITE** |  | |
| **4** | | **SYSTEM DESIGN** |  | |
| **4.1** | | **INTRODUCTION** |  | |
| **4.2** | | **UML DIAGRAM** |  | |
| **4.2.1** | | **USE CASE DIAGRAM** |  | |
| **4.2.2** | | **SEQUENCE DIAGRAM** |  | |
| **4.2.3** | | **STATE CHART DIAGRAM** |  | |
| **4.2.4** | | **ACTIVITY DIAGRAM** |  | |
| **4.2.5** | | **CLASS DIAGRAM** |  | |
| **4.2.6** | | **OBJECT DIAGRAM** |  | |
| **4.2.7** | | **COMPONENT DIAGRAM** |  | |
| **4.2.8** | | **DEPLOYMENT DIAGRAM** |  | |
| **4.2.9** | | **COLLABORATION DIAGRAM** |  | |
| **4.3** | | **USER INTERFACE DESIGN USING FIGMA** |  | |
| **4.4** | | **DATABASE DESIGN** |  | |
| **5** | | **SYSTEM TESTING** |  | |
| **5.1** | | **INTRODUCTION** |  | |
| **5.2** | | **TEST PLAN** |  | |
| **5.2.1** | **UNIT TESTING** | |  |
| **5.2.2** | **INTEGRATION TESTING** | |  |
| **5.2.3** | **VALIDATION TESTING** | |  |
| **5.2.4** | **USER ACCEPTANCE TESTING** | |  |
| **5.2.5** | **AUTOMATION TESTING** | |  |
| **5.2.6** | **SELENIUM TESTING** | |  |
| **6** | **IMPLEMENTATION** | |  |
| **6.1** | **INTRODUCTION** | |  |
| **6.2** | **IMPLEMENTATION PROCEDURE** | |  |
| **6.2.1** | **USER TRAINING** | |  |
| **6.2.2** | **TRAINING ON APPLICATION SOFTWARE** | |  |
| **6.2.3** | **SYSTEM MAINTENANCE** | |  |
| **7** | **CONCLUSION & FUTURE SCOPE** | |  |
| **7.1** | **CONCLUSION** | |  |
| **7.2** | **FUTURE SCOPE** | |  |
| **8** | **BIBLIOGRAPHY** | |  |
| **9** | **APPENDIX** | |  |
| **9.1** | **SAMPLE CODE** | |  |
| **9.2** | **SCREEN SHOTS** | |  |
| **9.2** | **GIT LOG** | |  |

## List of Abbreviations

IDE - Integrated Development Environment

HTML - Hyper Text Markup Language

CSS - Cascading Style Sheet

SQL - Structured Query Language

UML - Unified Modeling Language

PHP - Hypertext Preprocessor

# CHAPTER 1

# INTRODUCTION

### PROJECT OVERVIEW

**NurtureNest** is an advanced web-based platform designed to help parents manage their baby's health and developmental milestones with ease. It offers a range of features, including vaccination scheduling, personalized feeding suggestions, mental health tips, and the ability to maintain comprehensive health records for children. With a user-friendly interface, parents can create profiles, book vaccination appointments, and receive timely reminders for upcoming check-ups. The platform also enables secure communication between parents and healthcare providers for consultations and follow-ups. NurtureNest automates tasks associated with monitoring a child’s health, making it easier for both parents and healthcare providers to manage health records, vaccination schedules, and growth tracking. Administrators efficiently handle vaccine inventory, update nutritional guidelines, and approve health center requests, ensuring smooth operations. Available 24/7, NurtureNest provides easy access to vital health information anytime, ensuring convenience and reliability. Developed with Python Django for the backend, and HTML, CSS, JavaScript, AJAX, and jQuery for the frontend, the platform uses a robust SQLite database for efficient data management.

### PROJECT SPECIFICATION

The proposed system, NurtureNest, is a comprehensive web-based platform designed to help parents manage their baby’s health and developmental milestones with ease. The platform automates various health tracking and management tasks, making it user-friendly and available 24/7. Registered parents can create profiles for their children, book vaccination appointments, receive timely reminders, and access personalized feeding suggestions and mental health tips from the comfort of their homes. Parents and healthcare providers can securely communicate and manage health records, while admins oversee platform operations. The system consists of three main modules:

**Admin Module:** The admin has full control over the platform, managing vaccination schedules, feeding charts, and mental health tips. Admins can approve or reject health center requests, oversee vaccine inventory, and manage user profiles, ensuring smooth platform operations. They can also generate reports on vaccine requests, stock levels, and user activity.

* Login
* Manage vaccine schedules
* Manage feeding suggestions
* Add, edit, or delete mental health tips
* Approve or reject health centre profiles
* Oversee vaccine stock
* View and manage user profiles (parents and health centers)
* Generate reports on vaccine requests and inventory
* Edit profile
* Logout

**Parent Module:** Parents can register on the platform, complete their child’s profile, and schedule vaccination appointments. They receive reminders for upcoming appointments and health check-ups. Parents can also view feeding charts and mental health tips tailored to their child’s age, and securely access or update their child's health records. The platform allows parents to communicate with healthcare providers and track their child’s growth and development.

* Registration
* Login
* Forgot Password
* Change Password
* Complete and edit child profile
* Schedule vaccinations
* Receive reminders and notifications
* View feeding charts and mental health tips
* View vaccination records and health history
* Secure communication with healthcare providers
* Logout

**Healthcare Provider Module:** Healthcare providers, such as health centers, can manage vaccine stock, approve or reschedule vaccination appointments, and request vaccines from the admin. Providers can also upload vaccine images for machine-learning-assisted documentation, ensuring all relevant information is automatically extracted. Providers only gain access after admin approval and can then manage their profiles, appointments, and stock efficiently.

* Login
* Complete health center profile
* Manage vaccine appointments
* Request vaccine stock from admin
* Upload vaccine images for automated documentation
* Manage and view vaccination schedules
* Edit profile
* Logout

# CHAPTER 2

# SYSTEM STUDY

# 

### INTRODUCTION

System study for NurtureNest involves a thorough analysis of the existing baby health management processes to identify potential areas for improvement. This analysis focuses on understanding how parents, healthcare providers, and administrators currently manage baby health records, vaccination schedules, and developmental milestones, highlighting strengths and pinpointing inefficiencies in the existing methods. The primary goal of the system study is to evaluate the current processes and propose changes that would make health tracking more streamlined, user-friendly, and effective for both parents and medical professionals.

The results of this system study will be used to guide the development and optimization of NurtureNest, ensuring it meets user needs more effectively by automating key tasks such as vaccination scheduling, reminders, and health record management. By improving the platform's functionality, NurtureNest can deliver better outcomes in terms of parental ease-of-use, timely healthcare delivery, and optimized communication between parents and healthcare providers.

Various techniques are used to conduct the system study, including user interviews, surveys, document analysis, and data evaluation. Interviews with parents and healthcare providers help gather valuable insights into user needs and challenges. Document analysis offers an understanding of how the current system is managed, while data analysis reveals trends in child health monitoring and vaccination adherence. This comprehensive analysis ensures that NurtureNest is equipped to handle all aspects of child health management, from vaccination reminders to growth tracking.

The preliminary study in this system study is crucial in determining the feasibility of NurtureNest. This step involves assessing whether NurtureNest can effectively address the identified challenges in child healthcare management, ensuring that the platform is designed to align with the goals of enhancing parental engagement, simplifying health record management, and improving communication with healthcare providers.

### LITERATURE REVIEW

The Vaccine Image Classification and Details Retrieval System leverages machine learning and image processing techniques, widely studied for their effectiveness in medical image classification. Support Vector Machines (SVMs) are commonly chosen for tasks involving high-dimensional data due to their robustness and accuracy. Studies show that preprocessing steps like grayscale conversion and resizing improve model performance by standardizing image inputs and reducing noise.

Research also emphasizes the importance of structured datasets containing metadata, such as recommended age ranges, vaccine purpose, and potential side effects. This additional information supports comprehensive retrieval systems that can aid clinical decision-making, enhancing patient care by providing relevant vaccine information alongside classifications.

By drawing from these findings, the system achieves accurate classification and detail retrieval, combining proven techniques in machine learning and data structuring to deliver accessible, reliable vaccine information.

### PROPOSED SYSTEM

The proposed system, NurtureNest, addresses the limitations of the existing manual health management system by offering a comprehensive, digital platform for parents and healthcare providers. NurtureNest provides a user-friendly interface that allows parents to easily manage their child's health records, vaccination schedules, and growth tracking all in one place. The system supports real-time communication between parents and healthcare providers, enabling remote consultations, follow-ups, and timely advice without the need for in-person visits.

The admin can efficiently manage user accounts, update vaccination inventories, and approve health centre requests, ensuring smooth operation of the system. By automating the tracking and management of health information, NurtureNest reduces the need for manual data entry and lowers the risk of errors. It also stores health records securely in a centralized database, making the data easily accessible to authorized users while ensuring data privacy and security.

Parents can schedule vaccination appointments, receive reminders, and track their child's developmental milestones from the convenience of their home. NurtureNest is available 24/7, allowing parents and healthcare providers to access critical health data anytime, enhancing the overall efficiency of child health management.

### ADVANTAGES OF PROPOSED SYSTEM

* Available 24 hours a day, providing continuous access to health information.
* Saves time and effort by reducing the need for physical visits to healthcare centers.
* Easy to schedule appointments and track vaccinations.
* Secure online communication between parents and healthcare providers.
* User-friendly interface for managing health records and child growth tracking.
* Comprehensive health information, including vaccination schedules and developmental milestones.
* Accessible from anywhere with an internet connection.
* Automated reminders for upcoming vaccinations and health check-ups.
* Secure storage and management of health data

# CHAPTER 3

# REQUIREMENT ANALYSIS

## FEASIBILITY STUDY

Feasibility study is an important phase in software development process. A feasibility study is a way of figuring out if a proposed project or business idea is possible and if it makes sense financially. It involves looking at things like the size of the market, the competition, the costs involved, and whether the project can comply with laws and regulations. It also looks at whether the project can be carried out from a technical and operational standpoint. The study helps to identify risks and challenges that may come up and helps to determine whether the project is worth pursuing.

A feasibility study of a system proposal is according to its workability, which is the impact on the organization, ability to meet their user needs and effective use of resources. Thus, when a new application is proposed it normally goes through a feasibility study before it is approved for development.

The document provides the feasibility of the project that is being designed and lists various areas that were considered very carefully during the feasibility study of this project such as Technical, Economic and Operational feasibilities. The objective of the feasibility study is to establish the reasons for developing the software that is acceptable to users, adaptable to change and conformable to established standard.

### Economical Feasibility

Economic feasibility determines whether the required software is capable of generating financial gains for an organization. The developing system must be justified by cost and benefit.

NurtureNest is economically feasible. The cost of developing NurtureNest was carefully managed by utilizing open-source software and technology, significantly reducing overall expenses. The project’s cost was divided into development, system setup, and hosting. Since it was built using Python Django, an open-source framework, and SQLite for data management, the system was developed at a low cost. When compared to the benefits, including improved baby health monitoring and enhanced user convenience, the investment in NurtureNest is justified. The system is also designed to allow for future enhancements and updates without major financial impacts.

NurtureNest offers the following benefits:

* Reduces the time and effort required for tracking baby health and developmental milestones.
* Reduces manual work for healthcare providers and parents by automating health tracking.
* Enables efficient administration, providing healthcare providers with easy access to patient data.
* The platform is user-friendly, parent-focused, and accessible, enhancing the overall user experience.

### Technical Feasibility

Technical feasibility refers to the ability of a proposed project or business to be developed using available technology and resources. It involves evaluating whether the proposed project can be implemented using current technologies, infrastructure, and resources, and whether the project can be developed and maintained within the available budget and timeframe.

NurtureNest is technically feasible due to the use of modern and reliable technologies. The platform is developed using Python Django for the backend, which provides robust support for web development, ensuring scalability and maintainability. The SQLite database is used for efficient data management, offering quick access to health records and vaccination schedules. For the frontend, technologies such as HTML, CSS, JavaScript, jQuery, AJAX, and Bootstrap are utilized to create a responsive and user-friendly interface that works smoothly across devices. Additionally, NurtureNest integrates machine learning algorithms to provide personalized insights and predictive analytics for child health and growth tracking. The system was developed on a machine equipped with an Intel i7 core processor, 16GB of RAM, and SSD storage, ensuring that the project runs efficiently within the available technical resources

### Behavioral Feasibility

Behavioral feasibility refers to the ability of a proposed project or business to be accepted by its intended users or stakeholders. It involves evaluating whether the proposed project is likely to be used and accepted by the people it is intended for. This includes analyzing user behavior and preferences, cultural and social factors, as well as assessing potential resistance to change. NurtureNest is behaviorally feasible, designed to be well-accepted by its target users—parents, healthcare providers, and administrators. The platform addresses the need for efficient baby health monitoring and developmental tracking, saving users time and providing valuable insights, which makes it highly appealing. By shifting from traditional manual methods to a digital solution, NurtureNest enhances both accuracy and convenience. It is specifically tailored for managing baby health, offering accurate and well-organized information for decision-making. The platform is flexible, expandable, and accessible 24/7, ensuring users can access data at any time. Its user-friendly interface makes it easy to use, and the system is built to adapt to changes effortlessly. By reducing manual workloads and providing real-time insights, NurtureNest improves overall performance and ensures a smooth transition to the new system, making it likely to be widely accepted and adopted by its intended audience.

**3.1.4 Feasibility Study Questionnaire**

**Questionnaire for Parents:**

1. How do you currently keep track of your child's vaccination schedule?

We keep track of our child's vaccination schedule through reminders from the local health center, which calls to inform us of the date and time for our child's vaccinations.

1. What challenges do you face in remembering vaccination dates?

The main challenge we face in remembering vaccination dates is the lack of a consistent reminder system; we rely on calls from the local health centre and often mark the dates on a calendar ourselves.

1. How often do you visit your healthcare provider for your child's check-ups?

We do not visit the hospital weekly; we only go if our child has any health issues, whether minor or serious, and when the health centre informs us about scheduled check-ups.

1. What kind of reminders (email, SMS, app notifications) would you prefer for upcoming vaccinations?

We would prefer any of the above reminders—email, SMS, or app notifications—as all are convenient for us to keep track of upcoming vaccinations.

1. What information do you find most helpful in managing your child's growth and development?

We find information on developmental milestones, nutrition guidelines, vaccination schedules, and tips for managing common childhood illnesses most helpful in managing our child's growth and development.

1. How do you currently receive updates on your child’s growth milestones and nutritional needs?

We currently do not receive regular updates on our child's growth milestones and nutritional needs; we mainly rely on advice from our parents or grandparents and refer to a short handbook provided by the health centre for information on vaccinations and growth.

1. What features would you like to see in a baby health management platform?

We would like a baby health management platform that includes features such as receiving updates about vaccinations, scheduling vaccinations conveniently, providing instructions on baby care during different growth stages, and offering guidance on suitable activities and foods for optimal growth and development.

1. Are you interested in virtual consultations with your healthcare provider? Why or why not?

Yes, we are interested in virtual consultations with our healthcare provider because they offer convenience, reduce travel time, and allow us to seek medical advice from the comfort of our home, which is especially helpful during busy schedules or when facing minor health concerns.

1. How do you usually find activities and exercises suitable for your child's developmental stage?

We typically find activities and exercises suitable for our child's developmental stage through advice from parents or grandparents, as well as from information provided by the health center in the handbook given during vaccination appointments.

**Questions for Healthcare Providers (about Vaccination Maintenance):**

1. What system do you currently use to manage patient vaccination records?

We use the U-WIN system to manage patient vaccination records, which facilitates accurate tracking, updating, and timely reminders for vaccinations.

1. How do you remind parents about upcoming vaccination appointments?

We remind parents about upcoming vaccination appointments through Short Message Service (SMS).

1. What are the common reasons for missed vaccination appointments in your practice?

The common reason for missed vaccination appointments in our practice is a lack of awareness among patients and their families.

1. How do you ensure that vaccination records are kept up-to-date?

We ensure that vaccination records are kept up-to-date using the Kiddohealth app.

1. What challenges do you face in maintaining accurate vaccination schedules for patients?

We face challenges in maintaining accurate vaccination schedules due to the incomplete understanding of how immunity develops among patients and their families.

1. How do you communicate vaccination schedules and updates to parents?

We communicate vaccination schedules and updates to parents through immunization cards.

1. What features would help you better manage and track patient vaccinations?

The features of the CARD system would help us better manage and track patient vaccinations.

1. How do you handle deviations from the recommended vaccination schedule?

We handle deviations from the recommended vaccination schedule by following the ACIP immunization schedule guidelines and providing personalized recommendations based on individual patient needs.

1. How do you monitor vaccine stock levels and manage them?

The Min/Max inventory control system is used to monitor vaccine stock levels and manage them efficiently in our hospital.

1. What procedures are in place for ordering new vaccines when stock is low?

Procedures for ordering new vaccines when stock is low include reviewing past ordering records to inform timely replenishment decisions in our hospital.

## SYSTEM SPECIFICATION

### Hardware Specification

Processor - Intel Corei51135G7

RAM - 512GB

Hard disk - SSD

### Software Specification

Front End - HTML, CSS

Back End - Python Django

Database - SQLite

Client on PC - Windows 7 and above.

Technologies used - JS, HTML5, AJAX, J Query, Python Django, CSS, Machine Learning

## SOFTWARE DESCRIPTION

### Eg. Python Django

Python Django is a high-level web framework that enables rapid development of secure and maintainable websites. Created in 2005 by Adrian Holovaty and Simon Willison, Django is built using the Python programming language and follows the model-template-view (MTV) architectural pattern. As of 2024, Django remains one of the most popular frameworks for building robust, scalable web applications. Django emphasizes reusability, less code, and the "don't repeat yourself" (DRY) principle, which allows developers to focus on writing efficient code. It comes with a built-in admin interface, form handling, user authentication, and security features like protection against cross-site request forgery (CSRF), SQL injection, and cross-site scripting (XSS).

Django is an open-source framework and is known for its simplicity and speed of development, making it suitable for both small projects and large-scale enterprise applications. It works seamlessly with various databases, including PostgreSQL, MySQL, SQLite, and Oracle, allowing developers to create data-driven applications efficiently. With Django’s object-relational mapping (ORM) system, developers can interact with the database using Python code instead of SQL, further simplifying database management. Its scalability ensures that web applications can handle high traffic loads with ease, making Django a top choice for building complex websites and applications. Additionally, Django supports the integration of external libraries and APIs, which enhances the framework's flexibility and power for web development.

### E.g. SQLite

SQLite is a lightweight, open-source relational database management system (RDBMS) that is widely used for managing data in embedded applications and smaller projects. First released in 2000, SQLite is known for its simplicity, portability, and self-contained nature, as it does not require a separate server to run. This makes it an ideal choice for applications that need a database engine but don't require the complexity of larger systems like MySQL or PostgreSQL. SQLite stores all its data in a single file, which makes it easy to set up, manage, and distribute across different environments.

One of the key features of SQLite is its small footprint and efficiency, allowing it to be embedded in mobile apps, desktop applications, browsers, and even IoT devices. Despite being lightweight, SQLite supports most of the SQL standards, including complex queries, transactions, and indexing, making it suitable for a wide variety of applications. Its simplicity makes it easy to integrate with programming languages like Python, Java, C, and PHP, and it is often used in frameworks like Django for development purposes.

SQLite is known for its reliability and stability, even in environments with limited resources. It does not require any configuration or management of a server, making it highly portable across different operating systems, including Windows, Linux, and macOS. Additionally, SQLite is ACID-compliant, ensuring that database transactions are reliable and secure. However, SQLite is best suited for applications with moderate data and traffic requirements, as it is not optimized for high-concurrency or large-scale distributed systems.

Overall, SQLite's simplicity, portability, and ease of use make it an ideal choice for small to medium-scale applications, mobile development, and situations where a full-fledged RDBMS may be overkill.

# CHAPTER 4

# SYSTEM DESIGN

* 1. **INTRODUCTION**

System design is a critical first step in the development phase of any software project. It involves defining the architecture of the system and identifying the major components that will be required to achieve the desired functionality. This process lays the foundation for the entire development process and helps ensure that the final product is both functional and maintainable. The first step in system design is to identify the goals and requirements of the system. This involves understanding the needs of the users and stakeholders, and determining what features and functionality the system should have. It is important to have a clear understanding of the goals and requirements upfront, as this will help ensure that the system is designed to meet these requirements.

Once the requirements have been identified, the next step is to create a high-level architecture for the system. This involves identifying the major components of the system, such as the user interface, database, and application logic. The high-level architecture should provide a clear picture of how these components will work together to achieve the desired functionality. The design phase of software development is a critical transition point where the user-oriented document is transformed into a more technical document that can be used by programmers and database personnel. The purpose of the design phase is to provide a detailed plan for how the software will be developed and implemented. System design goes through two phases of development: Logical and Physical Design.

## UML DIAGRAM

Unified Modeling Language (UML) diagrams are a visual representation of software systems that help developers to better understand and communicate the structure and behavior of the system. It is managed by the Object Management Group (OMG), a non-profit organization that oversees the development of software standards. UML is a standardized language used in software engineering and consists of a set of graphical notations that can be used to create diagrams that represent different aspects of a software system. UML diagrams can be used throughout the software development process, from the initial planning and requirements gathering phase to the final testing and deployment phases. They can help to ensure that everyone involved in the development process has a clear and consistent understanding of the system, and can communicate their ideas effectively.

UML diagrams are interconnected, which means that changes made to one diagram can affect other diagrams as well. This helps to ensure that the system is consistent and that all of its components work together as intended.

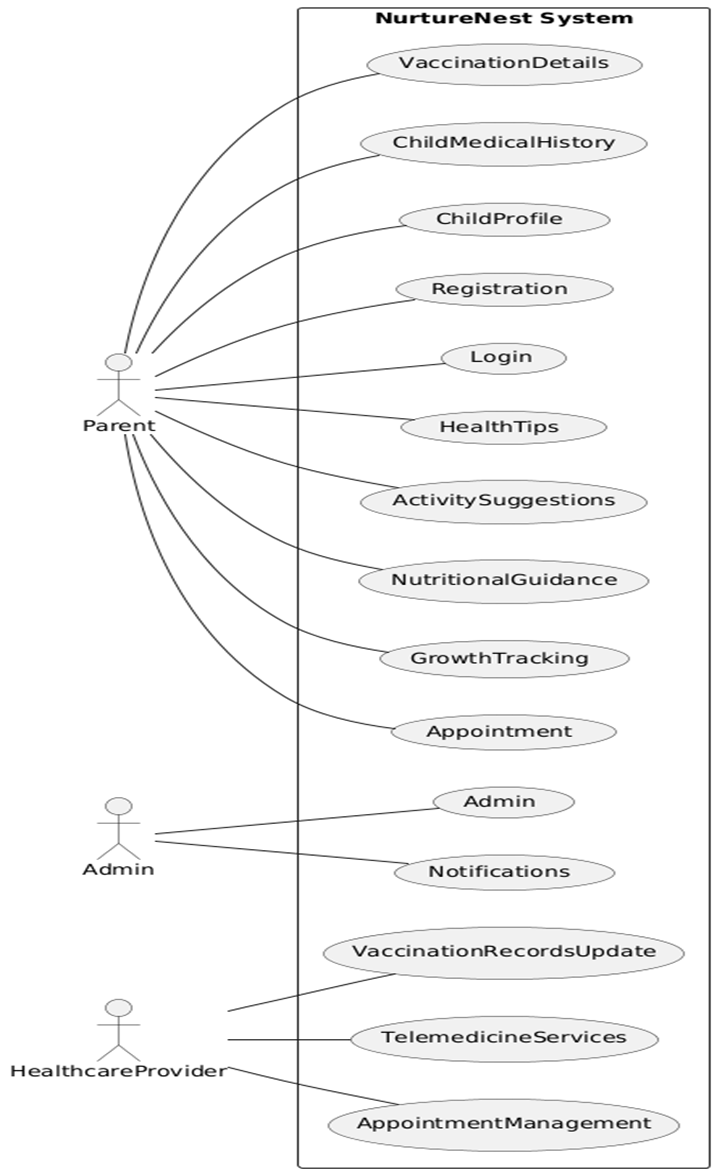
UML includes mainly following diagrams:

* Class Diagram
* Object Diagram
* Use Case Diagram
* Sequence Diagram
* Activity Diagram
* State chart Diagram
* Deployment Diagram
* Component Diagram
* Collaboration Diagram

## USE CASE DIAGRAM

A use case diagram is a visual representation of the interactions between actors (users or external systems) and the system being developed or analyzed. It is a powerful tool for requirements analysis as it helps to identify the different types of users and their goals, as well as the different functionalities that the system needs to provide. The use case diagram illustrates the different scenarios in which the system will be used, and it provides a high-level view of the system's behavior.

The use case diagram can be used in different stages of software development, from requirements gathering to testing and maintenance. It can help developers to communicate with stakeholders and to ensure that everyone has a shared understanding of the system's behavior. Use case diagrams can also be used to identify potential errors and edge cases that need to be considered during development. Finally, they can be used as a blueprint for testing, ensuring that all possible scenarios have been considered and that the system behaves as expected in each case.

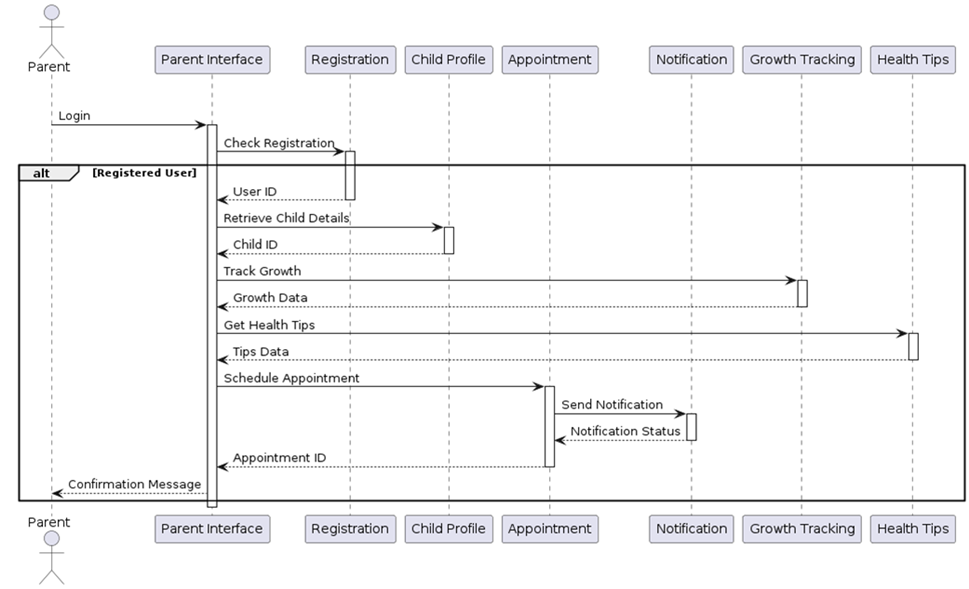


## SEQUENCE DIAGRAM

A sequence diagram is a graphical representation of the interactions that take place between objects or components in a system over time. It shows the order in which messages are sent between objects, as well as the timing of those messages. The vertical axis represents time, and the horizontal axis represents the objects or components involved in the interaction.

Each object or component is represented by a vertical line or "lifeline" that shows its lifespan during the interaction. Messages between objects are represented by arrows that point from the sender to the receiver, with the message content written above the arrow. The sequence diagram also shows any conditions or loops that affect the message flow, as well as any parallel interactions that occur.

Sequence diagrams are useful for visualizing the behavior of a system and identifying any potential issues or inefficiencies in its design. They can also help in the communication and collaboration between different teams working on the same system, as they provide a common language and understanding of how the system functions.



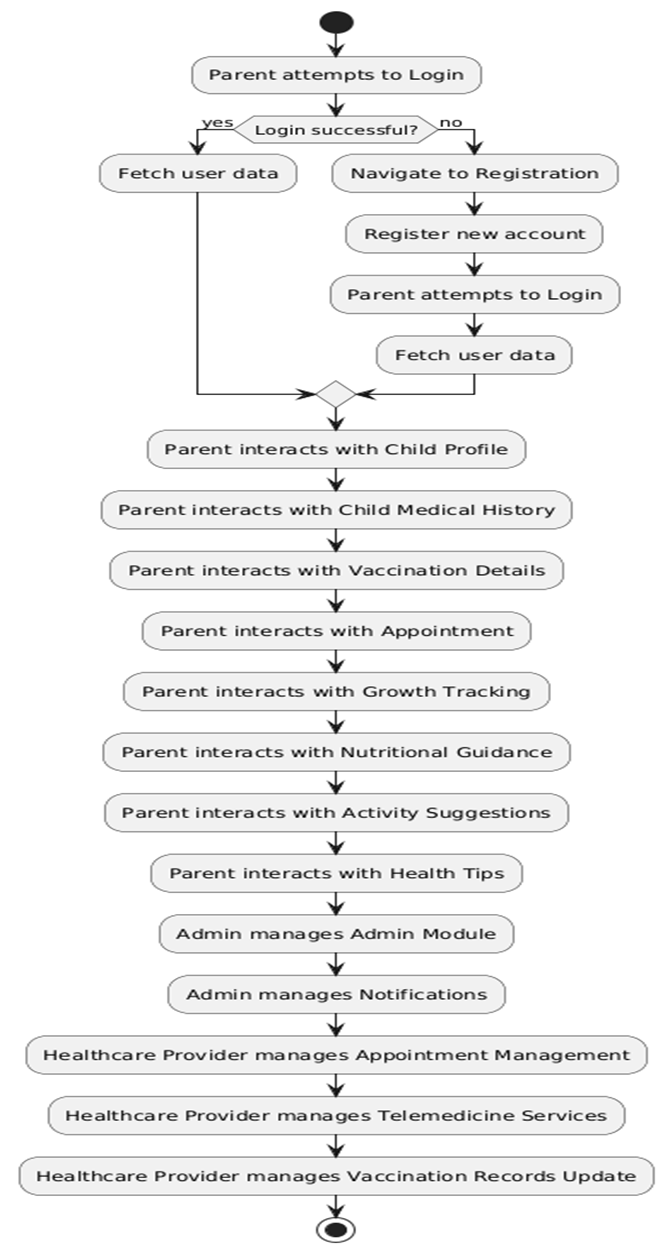
## 4.2.2 State Chart Diagram

Explanation, Diagram

## Activity Diagram

An activity diagram is a type of diagram used to represent the flow of activities or processes within a system. It shows the sequence of activities, the conditions for executing those activities, and the decisions that affect the flow of the activities. Activity diagrams are commonly used to model business processes, software systems, and other complex processes. In an activity diagram, activities are represented by nodes or boxes, and the flow between them is represented by arrows. The arrows may be labeled with conditions, decisions, or other factors that affect the flow of the activities. The activity diagram also shows the start and end points of the process, as well as any loops, parallel activities, or exceptions.

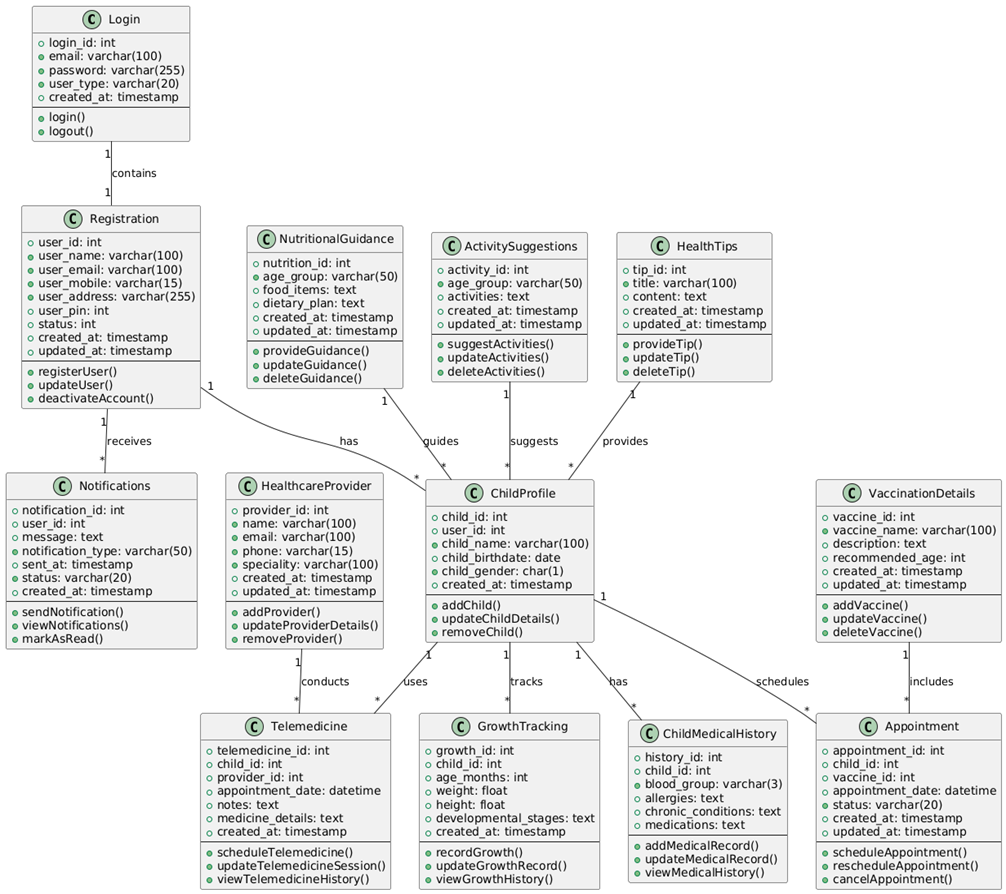
Activity diagrams are useful for visualizing complex processes and identifying any potential issues or inefficiencies in their design. They can help designers to understand the sequence of activities within a process, to optimize the process for efficiency and performance, and to identify any potential bottlenecks or areas for improvement. By breaking down a complex process into its individual activities and representing them in a clear and logical way, activity diagrams can help to improve communication and collaboration between different teams and stakeholders involved in the process.



## Class Diagram

A class diagram is a type of diagram used to represent the structure and relationships of objects in a system. It shows the classes, interfaces, attributes, and methods of the objects, as well as the relationships between them. Class diagrams are commonly used in object-oriented programming to model software systems. In a class diagram, classes are represented by rectangles with their names inside, and the attributes and methods of the classes are represented by smaller rectangles inside the larger rectangle. Relationships between classes are represented by lines between the rectangles, with different types of lines representing different types of relationships, such as inheritance, aggregation, or association.

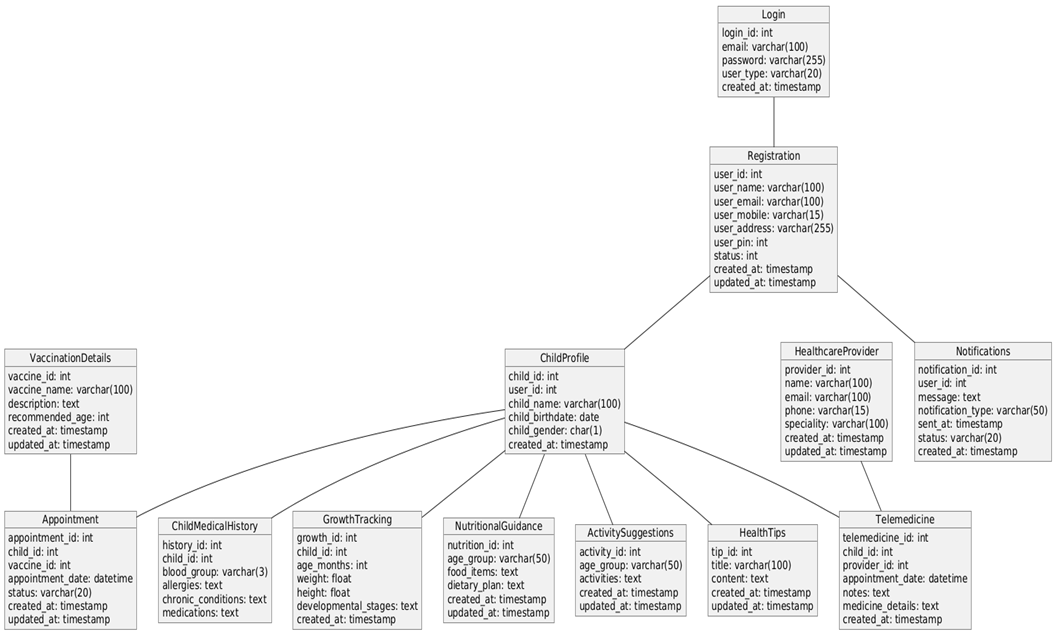
Class diagrams are useful for designing and understanding the structure of a system, as they provide a visual representation of the objects and their relationships. They can help designers to identify the different objects and their attributes and methods, and to ensure that the objects are organized in a logical and efficient way. By breaking down a complex system into its individual objects and representing them in a clear and logical way, class diagrams can help to improve communication and collaboration between different teams and stakeholders involved in the system.



## Object Diagram

An object diagram is a type of diagram used to represent a snapshot of the objects and their relationships in a system at a specific moment in time. It shows the objects, their attributes, and the relationships between them. Object diagrams are often used to illustrate the interactions between objects in a specific use case or scenario. In an object diagram, objects are represented by rectangles with their names inside, and the attributes and values of the objects are represented by smaller rectangles inside the larger rectangle. The relationships between objects are represented by lines between the rectangles, with different types of lines representing different types of relationships, such as composition, aggregation, or association.

Object diagrams are useful for understanding the behavior of a system in a specific use case or scenario. They can help designers to identify the objects and their attributes and values, and to ensure that the objects are organized in a logical and efficient way. By representing the objects and their relationships in a clear and simple way, object diagrams can help to improve communication and collaboration between different teams and stakeholders involved in the system.



## Component Diagram

Explanation, Diagram

**4.2.8 Deployment Diagram**

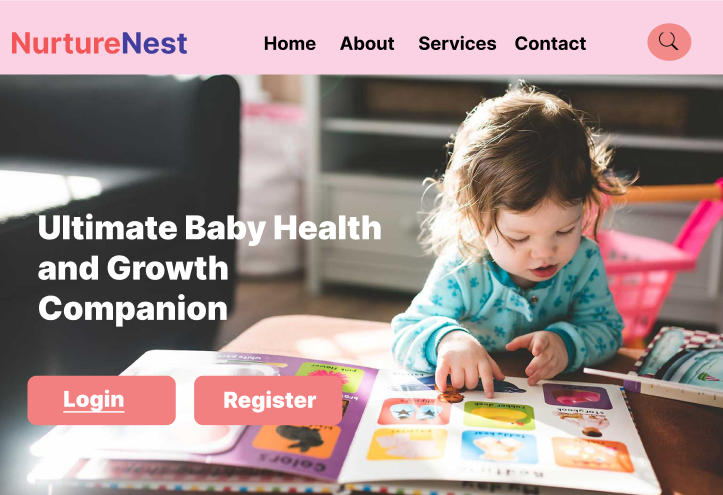
Explanation, Diagram

**4.2.9 Collaboration Diagram**

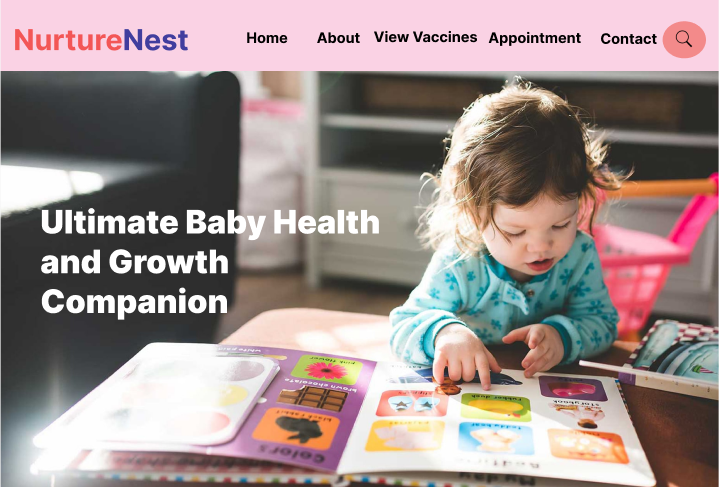
Explanation, Diagram

## 4.3 USER INTERFACE DESIGN USING FIGMA

**Form Name: Index Page**



**Form Name: User Home Page**

****

**Form Name: View Vaccines Page**

****

## 4.4 DATABASE DESIGN

Database design is the process of creating a data model for a database that accurately represents the entities, attributes, and relationships between data elements. The goal of database design is to create a database structure that is efficient, scalable, and easy to maintain. The process involves defining the database schema, selecting appropriate data types, and establishing relationships between tables. It also involves normalization, which is the process of minimizing redundancy in the database and ensuring that each piece of data is stored in only one place.

A well-designed database is essential for efficient data management, as it allows users to quickly and easily retrieve and manipulate data. The design must also consider security and access control, ensuring that sensitive data is protected and that users have appropriate levels of access to data. A poorly designed database can result in slow queries, data inconsistencies, and security vulnerabilities, which can be costly and time-consuming to fix. Therefore, it is important to carefully plan and design a database to ensure that it meets the needs of the organization and its users.

### 4.4.1 Relational Database Management System (RDBMS)

A Relational Database Management System (RDBMS) is a type of software used to manage and store data in a structured way. It relies on a relational model to organize data into tables, each of which contains related data that can be queried and manipulated using SQL (Structured Query Language). RDBMSs are used to manage large volumes of data and are commonly used in businesses, organizations, and governments to store and access important information.

One of the main advantages of using an RDBMS is its ability to ensure data consistency and accuracy. By enforcing constraints and relationships between tables, an RDBMS can prevent data inconsistencies and errors that can arise when data is stored in a non-standardized way. Another advantage is the ability to scale up as the amount of data grows. RDBMSs are designed to handle large volumes of data and can be scaled to accommodate the needs of growing businesses and organizations.

### 4.4.2 Normalization

Normalization is the process of organizing data in a relational database in order to minimize redundancy and dependency. It involves breaking down a larger table into smaller tables and establishing relationships between them. Normalization is important because it helps to reduce data redundancy, which can lead to inconsistencies and errors in the database.

There are several levels of normalization, referred to as normal forms. The most commonly used are first normal form (1NF), second normal form (2NF), and third normal form (3NF). Each level of normalization has specific requirements that must be met in order to achieve that level.

Normalization helps to ensure that data is stored in a consistent and logical manner, which makes it easier to retrieve and manipulate the data. It also reduces the amount of storage space required and improves database performance by reducing the amount of data that needs to be processed. However, over-normalization can also lead to performance issues, so it is important to strike a balance between normalization and efficient data retrieval.

**First Normal Form**

First Normal Form (1NF) is a fundamental concept in relational database design that establishes a set of rules for organizing data in a tabular format. According to 1NF, a table is in its first normal form if it contains no repeating groups or arrays. This means that every field in a table must contain atomic (indivisible) values, and there should be no repeating groups of columns. In other words, each table cell should hold a single, indivisible value, and every row should have a unique identifier, or a primary key, which is used to distinguish it from other rows in the table.

Achieving first normal form is important because it allows for data to be efficiently and accurately queried, updated, and maintained. It also eliminates data redundancy and inconsistencies, which can lead to errors and inefficiencies in the system. To ensure that a table is in 1NF, one should consider decomposing it into smaller tables that satisfy the atomicity and uniqueness requirements, and establishing relationships between those tables through primary and foreign keys. This process of normalization is essential for ensuring the integrity and efficiency of the database schema.

**Second Normal Form**

Second Normal Form (2NF) is a concept in relational database design that builds upon the principles of First Normal Form (1NF) by eliminating data redundancy and dependency issues in the database. According to 2NF, a table is in its second normal form if it is in 1NF and every non-key attribute is functionally dependent on the table's primary key. In other words, 2NF ensures that each table column is uniquely determined by the primary key, and there is no partial dependency between columns.

To achieve 2NF, one must identify and separate the columns that depend on a subset of the primary key and group them into a new table with a new primary key. This process is known as decomposition, and it helps eliminate data redundancy and inconsistency issues by breaking down tables into smaller, more manageable units. By decomposing the table into smaller units, it is possible to achieve higher levels of normalization, leading to more efficient and reliable database operations.

**Third Normal Form**

Third Normal Form (3NF) is another concept in relational database design that builds upon the principles of First Normal Form (1NF) and Second Normal Form (2NF) by further eliminating data redundancy and dependency issues. According to 3NF, a table is in its third normal form if it is in 2NF and there is no transitive dependency between non-key attributes. This means that every non-key attribute must be dependent only on the primary key, and not on any other non-key attribute.

To achieve 3NF, one must identify and remove any transitive dependencies that exist in the table. This can be achieved by decomposing the table into smaller units based on the functional dependencies of the attributes, and establishing relationships between these smaller units through foreign keys. By removing transitive dependencies, it is possible to achieve higher levels of normalization, leading to better data management and more efficient database operations.

Achieving 3NF is important because it helps ensure data integrity, consistency, and accuracy. By eliminating transitive dependencies, it reduces data redundancy and inconsistency issues, making the database more reliable and easier to maintain. Additionally, it simplifies the database design, making it more efficient and easier to use. Overall, 3NF is an important concept in relational database design that helps ensure the efficiency, reliability, and consistency of the database schema.

**Boyce Codd Normal Form**

Boyce-Codd Normal Form (BCNF) is a database normalization technique that helps to ensure that the database tables are well-structured and free from anomalies. The BCNF is a higher level of normalization compared to other normal forms such as first normal form (1NF), second normal form (2NF), and third normal form (3NF). BCNF is achieved by decomposing tables that contain functional dependencies that violate the Boyce-Codd normal form.

A table is considered to be in BCNF if it satisfies the following criteria:

* Every determinant in the table is a candidate key, which means that there are no non-trivial dependencies between attributes.
* The table must not contain any non-trivial functional dependencies between attributes that are not part of any candidate key.

If a table does not meet these criteria, it can be decomposed into smaller tables that are in BCNF. This helps to eliminate redundancies and anomalies in the database. However, it's important to note that decomposing tables can sometimes result in an increase in the number of tables and may negatively impact query performance.

**Fourth Normal Form**

Fourth Normal Form (4NF) is a further level of database normalization that addresses multi-valued dependencies between attributes. It is an extension of the third normal form (3NF) and aims to eliminate redundancy in multi-valued dependencies.

A multi-valued dependency occurs when there is a relationship between non-key attributes that is not fully dependent on the primary key. In other words, there are non-key attributes that depend on a combination of other non-key attributes, rather than on the primary key alone.

To achieve 4NF, we decompose tables that contain multi-valued dependencies into smaller tables, each of which contains a single multi-valued dependency. In this way, we can eliminate redundancies and ensure that the database is well-structured and free from anomalies.

It's important to note that 4NF is not always necessary for every database, and it may not always be feasible to decompose tables into smaller tables due to performance or other practical considerations. However, in cases where multi-valued dependencies exist, 4NF can be a useful technique to improve data quality and consistency.

### 4.4.3 Sanitization

Sanitization in the context of ensuring that no residual data can be recovered even after a comprehensive forensic investigation is known as secure data wiping or secure data erasure. This process involves permanently deleting data from a storage device, ensuring that it cannot be recovered even through advanced forensic techniques.

To ensure that data is securely wiped, various techniques and tools can be used, such as overwriting the data with random patterns multiple times or physically destroying the storage device. The exact method used depends on the sensitivity of the data and the level of security required.

Secure data wiping is essential in cases where sensitive or confidential data needs to be permanently erased, such as when disposing of old hardware, transferring ownership of a device, or closing down a business. Failure to properly sanitize data can result in the unintentional exposure of sensitive information, which can have serious legal, financial, or reputational consequences.

**4.4.4 Indexing**

In SQL, indexing is the process of creating data structures that allow for faster retrieval of data from a database. Indexes are essentially lists of keys that point to specific locations within a table, making it possible to quickly locate and retrieve data that matches specific search criteria.

Creating indexes in SQL can significantly improve query performance, especially when working with large databases. By creating indexes on frequently accessed columns, such as primary keys or foreign keys, SQL can quickly locate the data that matches specific search criteria, without having to scan the entire table.

It's important to note that indexing can also have a negative impact on performance if done incorrectly. Over-indexing, or creating too many indexes, can slow down the insertion and updating of data, as well as increase the size of the database. Therefore, it's important to carefully consider the columns to index and the type of index to use to ensure that the indexing process does not adversely affect overall database performance.

### 4.5 TABLE DESIGN

**1.tbl\_user**

Primary key: user\_id

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No: | Fieldname | Datatype (Size) | Key Constraints | Description of the Field |
| 1 | user\_id | INT (11) | PRIMARY KEY | Primary Key of tbl\_login |
| 2 | First\_Name | VARCHAR (50) |  | To store the first\_name of the users |
| 3 | Last\_named | VARCHAR (50) |  | To store the last\_name of the users |
| 4 | email | VARCHAR (50) |  | To store the email of the users |
| 5 | Password | VARCHAR (50) |  | To store the password of the users |
| 6 | User\_type | VARCHAR (30) |  | To store the user type of the loginned users |
| 7 | status | VARCHAR (10) |  | To store the status of the users who are active |
| 8 | Date\_joined | DATE (100) |  | To store the users joined date and time |

**2.tbl\_parentprofile**

Primary key: parent\_id

Foreign Key: user\_id references table tbl\_user

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No: | Fieldname | Datatype (Size) | Key Constraints | Description of the Field |
| 1 | parent \_id | Int (11) | Primary Key | Primary Key of tbl\_ parentprofile |
|  | user\_id | Int (11) | Foreign Key | Primary Key of tbl\_user |
| 2 | contact\_no | varchar (50) |  | To store the mobile number of the parent |
| 3 | address | varchar (50) |  | To store the address the parent |
| 4 | place | varchar (50) |  | To store the place details of the parent |
| 5 | parentno | varchar (20) |  | To store the unique number that is given to the parents |

**3. tbl\_childprofile**

Primary Key: child\_id

Foreign Key: parent \_id references table tbl\_ parentprofile

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No: | Fieldname | Datatype (Size) | Key Constraints | Description of the Field |
| 1 | child\_id | Int (11) | Primary Key | Primary key of tbl\_childprofile |
| 2 | parent \_id | Int (11) | Foreign Key | Primary Key of tbl\_ parentprofile |
| 3 | child\_name | VARCHAR (100) | NOT NULL | To store the Child's name |
| 4 | child\_birthdate | DATE | NOT NULL | To store the Child's birthdate |
| 5 | child\_gender | CHAR (1) | NOT NULL | To store the Child's gender (M/F) |
| 6 | Bloog\_group | CHAR (5) | NOT NULL | To store the Child’s blood group |
| 7 | birth\_weight | Decimal (5) | NOT NULL | To store the birth\_weight of the child |
| 8 | birth\_height | Decimal (5) | NOT NULL | To store the birth\_height of the child |
| 9 | current\_weight | Decimal (5) | NOT NULL | To store the current\_weight of the child |
| 10 | current\_height | Decimal (5) | NOT NULL | To store the current\_height of the child |
| 11 | age | CHAR (5) | NOT NULL | To store the age of the child |

**4.tbl\_** **VaccinationRecord**

Primary Key: vaccinationrecord\_id

Foreign Key: child\_id references table tbl\_childprofile

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No: | Fieldname | Datatype (Size) | Key Constraints | Description of the Field |
| 1 | vaccinationrecord\_id | INT (11) | PRIMARY KEY | Primary Key of tbl\_ VaccinationRecord |
| 2 | child\_id | INT (11) | FOREIGN KEY | Primary key of tbl\_ VaccinationRecord |
| 3 | Vaccine\_taken | BOOLEAN | NOT NULL | To store if the vaccine is taken or not |
| 4 | Vaccine\_name | VARCHAR (50) | NOT NULL | To store the name of the vaccine |
| 5 | place | VARCHAR (50) | NOT NULL | To store the name of place where the child takes the latest vaccine |
| 6 | date | DATE | NOT NULL | To store the name of date in which the child takes the latest vaccine |
| 7 | weight | DECIMAL | NOT NULL | To store the weight of the child at the time of the latest vaccine taken |

**5.tbl\_healthprofile**

Primary Key: healthprofile\_id

Foreign Key: user\_id references table tbl\_user

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No: | Fieldname | Datatype (Size) | Key Constraints | Description of the Field |
| 1 | healthprofile\_id | INT (11) | Primary key | Primary key of tbl\_ healthprofile |
| 2 | user\_id | INT (11) | Foreign Key | Primary Key of tbl\_user |
| 3 | Healthcenter\_name | VARCHAR (50) | Primary Key | To store the name of the health center |
| 4 | phone | VARCHAR (100) |  | To store the phone number of the healthcenter |
| 5 | address | VARCHAR (100) |  | To store the address of the healthcenter |
| 6 | city | VARCHAR (50) |  | To store the city name of the healthcenter |
| 7 | license\_number | VARCHAR (50) |  | To store license\_number of the healthcenter |

**6.tbl\_Vaccine**

Primary Key: vaccine\_id

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No: | Fieldname | Datatype (Size) | Key Constraints | Description of the Field |
| 1 | vaccine\_id | Int (11) | Primary Key | Primary Key of tbl\_Vaccine |
| 2 | Vaccine\_name | VARCHAR (50) | NOT NULL | To store the name of the vaccine |
| 3 | manufacturer | VARCHAR (50) | NOT NULL | To store the name of the manufacturer of the vaccine |
| 4 | batch\_number | VARCHAR (20) | NOT NULL | To store the batch number of the vaccine |
| 5 | date\_manufacture | DATE | NOT NULL | To store the manufacturing date of the vaccine |
| 6 | expiry\_date | DATE | NOT NULL | To store the expiry date for the vaccine |
| 7 | age\_group | VARCHAR (20) | NOT NULL | To store the age group of the vaccine administered |
| 8 | indications | VARCHAR (50) | NOT NULL | To store the description about the vaccine |
| 9 | stock | INT (20) | NOT NULL | To store the total quantity available for the vaccine |
| 10 | free\_or\_paid | INT (20) |  | To store whether the vaccine is paid or not |

**7.tbl\_VaccineDose**

Primary Key: dose\_id

Foreign Key: vaccine\_id references table tbl\_vaccine

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No: | Fieldname | Datatype (Size) | Key Constraints | Description of the Field |
| 1 | dose\_id | Int (11) | Primary Key | Primary Key of tbl\_vaccineDose |
| 2 | Vaccine\_id | Int (11) | Foreign key | Primary key of tbl\_vaccine |
| 3 | dose\_number | Int (11) |  | To store the dose numbers of the vaccines added |
| 3 | Interval\_days | VARCHAR(10) |  | To store the interval days for the next vaccine to be taken |

**8.tbl\_vaccineRequest**

Primary Key: vaccinerequest\_id

Foreign Key: vaccine\_id references table tbl\_vaccine

healthprofile\_id id references table tbl\_ healthprofile

dose\_id references table tbl\_vaccineDose

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No: | Fieldname | Datatype (Size) | Key Constraints | Description of the Field |
| 1 | vaccinerequest\_id | INT (11) | Primary Key | Primary Key of tbl\_vaccineRequest |
| 2 | vaccine\_id | INT (11) | Foreign key | Primary key of tbl\_vaccine |
| 3 | healthprofile\_id | INT (11) | Foreign key | Primary key of the tbl\_ healthprofile |
| 4 | dose\_id | INT (11) | Foreign key | Primary key of tbl\_vaccineDose |
| 5 | requested\_stock | INT (20) |  | To store the stock of the requested vaccine |
| 6 | status | VARCHAR (10) |  | To store the status of the reuqsted vaccine |
| 7 | request\_date | DATE |  | To store the date at which the vaccine is requetsed |
| 8 | approved\_date | DATE |  | To store the date which the requested stock of the vaccine get approved |

**9.tbl\_****appointment**

Primary Key: appointment\_id

Foreign Key: user\_id references table tbl\_user

healthprofile\_id id references table tbl\_ healthprofile

vaccine\_id references table tbl\_vaccine

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No: | Fieldname | Datatype (Size) | Key Constraints | Description of the Field |
| 1 | appointment\_id | INT (11) | Primary Key | Primary Key of tbl\_appointment |
| 2 | user\_id | INT (11) | Foreign key | Primary key of the tbl\_user |
| 3 | healthprofile\_id | INT (100) | Foreign key | Primary key of the tbl\_ healthprofile tbl\_user |
| 4 | vaccine\_id | VARCHAR (50) | Foreign key | Primary key of the tbl\_vaccine |
| 5 | appointment\_date | DATE |  | To store the date of the appointment taken |
| 6 | appointment\_time | TIMESTAMP |  | To store the time of the appointment taken |
| 7 | status | VARCHAR (20) |  | To store the status of the appointment |
| 8 | approval\_date | DATE |  | To store the date at which the appointment get approved |

**10.tbl\_notification**

Primary Key: notification\_id

Foreign Key: user\_id references table tbl\_ user

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No: | Fieldname | Datatype (Size) | Key Constraints | Description of the Field |
| 1 | notification\_id | Int (11) | Primary key | Primary key of tbl\_notification |
| 2 | user\_id | Int (11) | Foreign Key | Primary Key of tbl\_user |
| 3 | message | VARCHAR (20) |  | To store the notification message |
| 4 | created\_at | DATE |  | To store the date at which the notification received |
| 5 | is\_read | BOOLEAN |  | To store the status if the notification is read or not |

**11.tbl\_** **FeedingChart**

Primary Key: feedingchart\_id

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No: | Fieldname | Datatype (Size) | Key Constraints | Description of the Field |
| 1 | feedingchart\_id | Int (11) | Primary Key | Primary Key of tbl\_ feedingchart |
| 2 | age | VARCHAR (11) |  | To store the age group that description belongs to |
| 2 | main\_heading | VARCHAR (50) |  | To store the first heading for the age group about the feeding |
| 4 | description | VARCHAR (50) |  | To store the detailed description for the age group about the feeding |
| 4 | created\_at | DATE |  | To store the date at which it is created |

**12.tbl\_MentalHealthDetails**

Primary Key: mentalHealthDetails\_id

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No: | Fieldname | Datatype (Size) | Key Constraints | Description of the Field |
| 1 | mentalHealthDetails\_id | Int (11) | Primary Key | Primary Key of tbl\_ mentalHealthDetails |
| 2 | age | Int (11) | Foreign key | To store the age group |
| 3 | image | VARCHAR (50) |  | To store the images |

**12.tbl\_MentalHealthDescriptions**

Primary Key: mentalHealthDescriptions\_id

Foreign Key: mentalHealthDetails\_id references table tbl\_ mentalHealthDetails

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No: | Fieldname | Datatype (Size) | Key Constraints | Description of the Field |
| 1 | mentalHealthDescriptions\_id | Int (11) | Primary Key | Primary Key of tbl\_ mentalHealthDescriptions |
| 2 | descriptions | VARCHAR (50) | Foreign key | To store the descriptions |

# CHAPTER 5

# SYSTEM TESTING

* 1. **INTRODUCTION**

System testing is an essential aspect of the software development process that ensures that the entire system or application meets the specified requirements and works as intended. In the project report, the section on system testing should provide an in-depth description of the testing process, including the scope of testing, the testing methodology used, and the tools employed.

It should also cover the different types of system testing performed, such as functional testing, performance testing, security testing, usability testing, and compatibility testing, along with the test scenarios and test cases used. Additionally, the report should include a summary of the test results, highlighting any issues or defects detected and how they were resolved.

## TEST PLAN

A test plan in system testing typically involves testing the entire software system as a whole, including all its features and functions. The objective of system testing is to verify that the software meets its specified requirements and is functioning as expected in its intended environment.

The test plan typically includes details such as the testing environment, the test strategy, the test objectives, the test schedule, and the test team responsibilities. It may also include the testing techniques, tools, and metrics to be used. The test plan should be reviewed and approved by all stakeholders, including the project manager, development team, and testing team, to ensure that everyone is on the same page regarding the testing approach.

The test plan should also provide information on the different levels of testing. These typically include:

* Unit Testing
* Integration Testing
* System Testing
* Acceptance Testing

### Unit Testing

Unit testing is a type of testing that is focused on verifying the functionality of individual code units or modules. A well-designed test plan for unit testing should outline the specific objectives of unit testing, the approach to be taken, and the resources that will be required to carry out the testing. Unit testing is typically carried out by the development team and is an essential part of the software development lifecycle.

Unit testing is an important part of the overall testing process, as it can help to identify defects early in the software development lifecycle when they are easier and less expensive to fix. By creating a detailed test plan for unit testing, the development team can ensure that all code units have been thoroughly tested and that the software meets the desired quality standards. This can ultimately help to improve the reliability, performance, and maintainability of the software.

### Integration Testing

Integration testing is a type of testing that focuses on testing the interaction between different modules or components of the software system to ensure that they are working together as expected. In system testing, integration testing is typically performed after unit testing, and it is focused on testing the entire system as a whole. The goal of integration testing is to detect and resolve any issues that arise from the interaction between different modules before the system is released to users.

Integration testing can help to identify defects that may not have been caught during unit testing, as it focuses on the interaction between modules rather than on individual code units. By testing the software system as a whole, integration testing can help to ensure that the system meets the desired quality standards, performs as expected, and meets the needs of its users.

### Validation Testing or System Testing

Validation testing, also known as system testing, is a critical part of software testing that is conducted to evaluate the entire system or application under test. The goal of this type of testing is to validate that the software system meets its intended requirements and works as expected.

Validation testing involves a series of tests designed to check the software system's functionality, reliability, usability, performance, and security. It is typically performed after integration testing and prior to user acceptance testing. The objective of system testing is to ensure that the software system meets the customer's expectations and requirements before it is released to production.

### Output Testing or User Acceptance Testing

Output testing, also known as user acceptance testing (UAT), is a type of testing that focuses on ensuring that the software system meets the needs and expectations of the end-users. The purpose of user acceptance testing is to validate that the software system is ready for release and to ensure that it meets all the functional and non-functional requirements specified by the customer.

User acceptance testing is typically performed by end-users, business stakeholders, or subject matter experts who validate the system's functionality, usability, and overall user experience. The test cases for UAT are designed to simulate real-world scenarios and to verify that the system behaves as expected in different use cases.

User acceptance testing is a critical part of the software development life cycle, as it helps to ensure that the software system meets the needs and expectations of the end-users. It can help to identify any issues or defects that may have been missed during earlier stages of testing, and ensure that the system is ready for release. By performing UAT, organizations can reduce the risk of user dissatisfaction, enhance the user experience, and increase customer satisfaction with the software system.

* + 1. **Automation Testing**

Automation testing is the process of using software tools to execute pre-scripted tests on a software application or system. The goal of automation testing is to improve the efficiency and accuracy of the testing process while reducing the time and cost involved in manual testing. Automation testing can be used for functional testing, regression testing, performance testing, and other types of software testing.

One of the main benefits of automation testing is that it can help to reduce the time and effort required for repetitive testing tasks. Automated tests can be executed more quickly and consistently than manual tests, and they can be run repeatedly without the need for human intervention. Automation testing can also help to identify defects and issues more quickly and accurately than manual testing, as it can perform a large number of tests in a short period of time.

* + 1. **Selenium Testing**

Selenium is an open-source automation testing tool used to automate web browsers. It supports a wide range of programming languages, including Java, Python, and Ruby, and can be used to test web applications on different platforms and browsers. Selenium is widely used for functional testing, regression testing, and other types of testing, and it can help to improve the efficiency and accuracy of the testing process.

One of the key features of Selenium is its ability to simulate user interactions with web applications. Selenium can automate actions such as clicking buttons, filling out forms, and navigating through web pages, allowing testers to validate the functionality and usability of web applications. Selenium can also be used to test the responsiveness and performance of web applications by simulating multiple users accessing the application simultaneously.

**Example:**

**Test Case 1**

**Code**

import pytest

from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.support.ui import WebDriverWait

from selenium.webdriver.support import expected\_conditions as EC

from selenium.webdriver.chrome.options import Options

from selenium.common.exceptions import TimeoutException

class TestParentLoginTest():

def setup\_method(self, method):

chrome\_options = Options()

chrome\_options.add\_argument('--ignore-certificate-errors')

chrome\_options.add\_argument('--ignore-ssl-errors')

self.driver = webdriver.Chrome(options=chrome\_options)

self.driver.set\_window\_size(1050, 728)

self.vars = {}

def teardown\_method(self, method):

self.driver.quit()

def test\_parent\_login\_test(self):

self.driver.get("http://127.0.0.1:8000/login/")

try:

email\_field = WebDriverWait(self.driver, 10).until(

EC.visibility\_of\_element\_located((By.ID, "email"))

)

email\_field.click()

email\_field.send\_keys("simisajan002@gmail.com")

password\_field = self.driver.find\_element(By.ID, "password")

password\_field.click()

password\_field.send\_keys("Simi@123")

self.driver.find\_element(By.CSS\_SELECTOR, ".btn").click()

WebDriverWait(self.driver, 10).until(

EC.url\_contains("home")

)

schedule\_appointment\_button = WebDriverWait(self.driver, 10).until(

EC.element\_to\_be\_clickable((By.LINK\_TEXT, "Schedule Appointment"))

)

schedule\_appointment\_button.click()

view\_appointments\_button = WebDriverWait(self.driver, 10).until(

EC.element\_to\_be\_clickable((By.LINK\_TEXT, "View Appointments"))

)

view\_appointments\_button.click()

home\_link = WebDriverWait(self.driver, 10).until(

EC.element\_to\_be\_clickable((By.LINK\_TEXT, "Home"))

)

home\_link.click()

except TimeoutException:

print("Timeout occurred. Some elements were not found in time.")

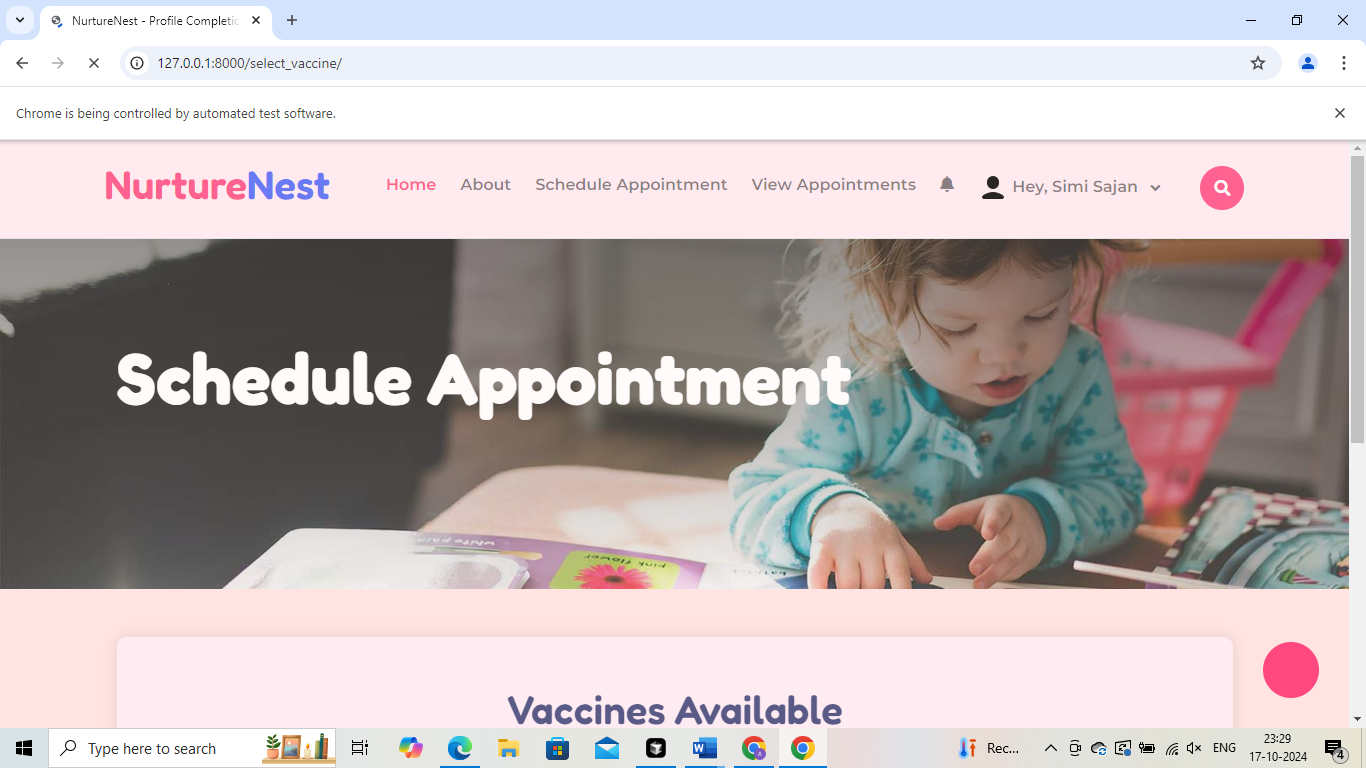
except Exception as e:

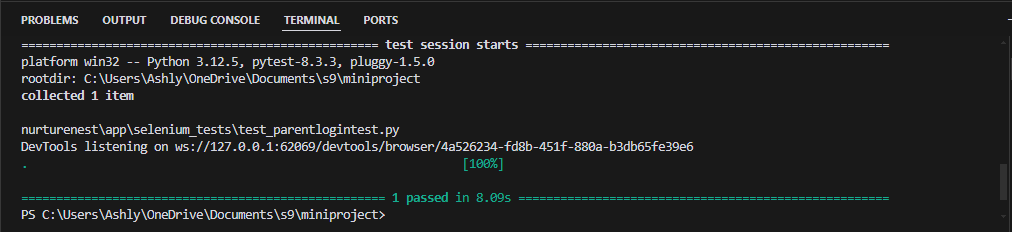
print(f"An error occurred: {e}")

if \_\_name\_\_ == "\_\_main\_\_":

pytest.main([\_\_file\_\_])

**Eg.Screenshot**





**Eg.Test Report**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case 1** | | | | | |
| **Project Name: NurtureNest** | | | | | |
| **Login Test Case of User** | | | | | |
| **Test Case ID: Test\_1** | | | **Test Designed By: Ashly Mol Thomas** | | |
| **Test Priority (Low/Medium/High): High** | | | **Test Designed Date: 17-10-2024** | | |
| **Module Name**: View User Page | | | **Test Executed By: Ms. Merin Mathew** | | |
| **Test Title: View user pages** | | | **Test Execution Date: 16-03-2023** | | |
| **Description: View the vaccine schedule and the user’s appointment details** | | |  | | |
| **Pre-Condition:** User has valid username and password and view their pages | | | | | |
| **Step** | **Test Step** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/**  **Fai l)** |
| 1 | Navigate to login page |  | Display Login page | Login page displayed | Pass |
| 2 | Valid username entered | Username:  simisajan@gmail.com | Display the Home page of the user | Home Page Displayed | Pass |
| 3 | Valid password entered | Simi@123 |
| 3 | Choose Schedule Appointment |  | View the vaccine page to select the vaccine | View the vaccine page to select the vaccine | Pass |
| **4** | Choose View Appointments |  | User can be able to view their scheduled appointment | User can be able to view their appointment | Pass |
| **Post-Condition: username and password is checked with database values and user can be able to schedule the appointment and view their appointments along with their current status** | | | | | |

**Test Case 2:**

**Code**

import pytest

from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.support.ui import WebDriverWait

from selenium.webdriver.support import expected\_conditions as EC

from selenium.webdriver.chrome.options import Options

from selenium.common.exceptions import TimeoutException

class TestAdminLoginTest:

def setup\_method(self, method):

chrome\_options = Options()

chrome\_options.add\_argument('--ignore-certificate-errors')

chrome\_options.add\_argument('--ignore-ssl-errors')

self.driver = webdriver.Chrome(options=chrome\_options)

self.driver.set\_window\_size(1382, 744)

self.vars = {}

def teardown\_method(self, method):

self.driver.quit()

def test\_admin\_login\_test(self):

self.driver.get("http://127.0.0.1:8000/login/")

try:

email\_field = WebDriverWait(self.driver, 10).until(

EC.visibility\_of\_element\_located((By.ID, "email"))

)

email\_field.click()

email\_field.send\_keys("nurturenest@gmail.com")

password\_field = self.driver.find\_element(By.ID, "password")

password\_field.click()

password\_field.send\_keys("Admin@123")

self.driver.find\_element(By.CSS\_SELECTOR, ".btn").click()

WebDriverWait(self.driver, 10).until(

EC.element\_to\_be\_clickable((By.LINK\_TEXT, "View Available Vaccines"))

).click()

WebDriverWait(self.driver, 10).until(

EC.element\_to\_be\_clickable((By.CSS\_SELECTOR, ".sidebar-link:nth-child(1)"))

).click()

WebDriverWait(self.driver, 10).until(

EC.element\_to\_be\_clickable((By.LINK\_TEXT, "Vaccines Requests"))

).click()

WebDriverWait(self.driver, 10).until(

EC.element\_to\_be\_clickable((By.LINK\_TEXT, "Dashboard"))

).click()

WebDriverWait(self.driver, 10).until(

EC.element\_to\_be\_clickable((By.LINK\_TEXT, "View Feeding Chart Details"))

).click()

WebDriverWait(self.driver, 10).until(

EC.element\_to\_be\_clickable((By.LINK\_TEXT, "Dashboard"))

).click()

profile\_icon = WebDriverWait(self.driver, 10).until(

EC.element\_to\_be\_clickable((By.CSS\_SELECTOR, ".profile-container"))

)

profile\_icon.click()

WebDriverWait(self.driver, 10).until(

EC.element\_to\_be\_clickable((By.LINK\_TEXT, "NurtureNest"))

).click()

WebDriverWait(self.driver, 10).until(

EC.element\_to\_be\_clickable((By.LINK\_TEXT, "Logout"))

).click()

except TimeoutException:

print("Timeout occurred. Some elements were not found in time.")

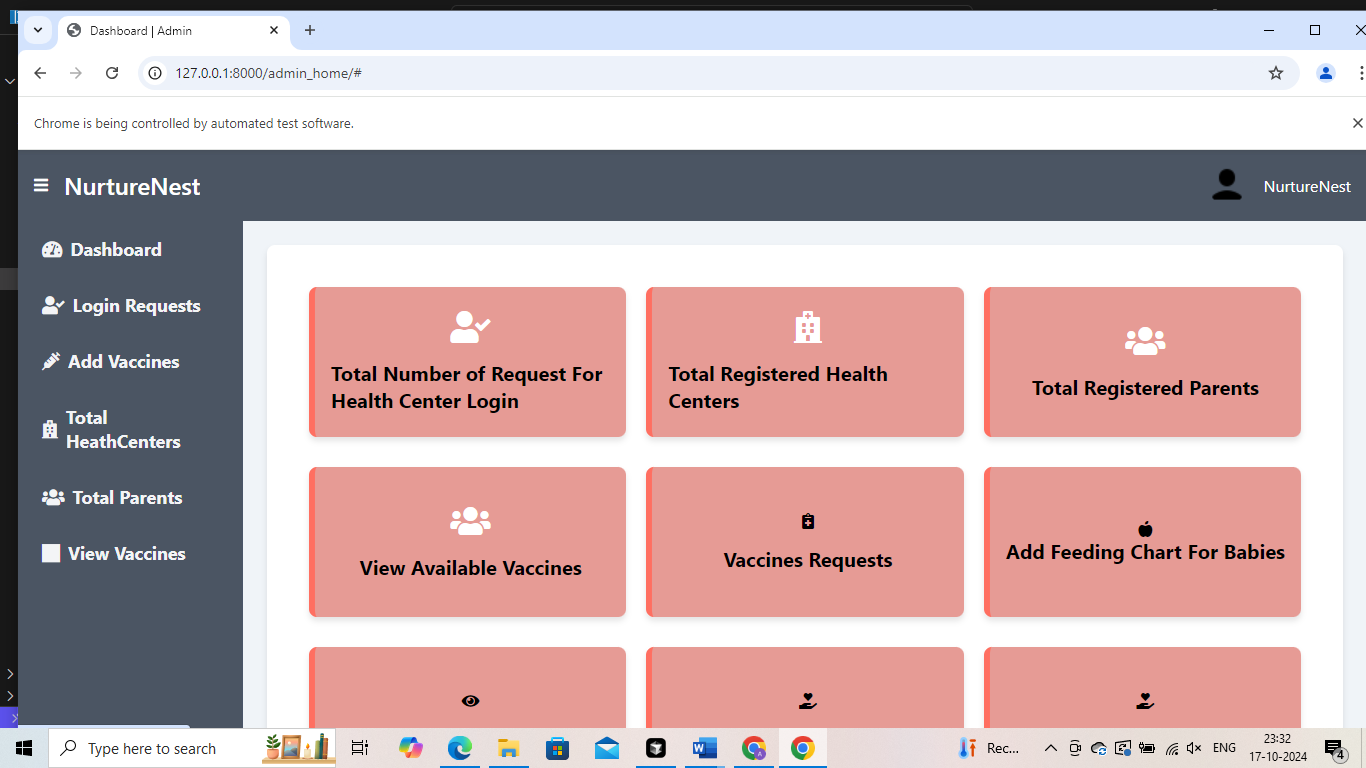
except Exception as e:

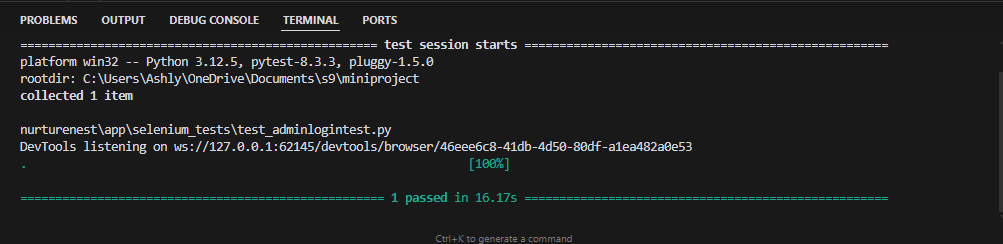
print(f"An error occurred: {e}")

if \_\_name\_\_ == "\_\_main\_\_":

pytest.main([\_\_file\_\_])

**Screenshot**

****

****

**Test report**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case 2** | | | | | |
| **Project Name: NurtureNest** | | | | | |
| **Login Test Case of Admin** | | | | | |
| **Test Case ID: Test\_2** | | | **Test Designed By: Ashlymol Thomas** | | |
| **Test Priority (Low/Medium/High): High** | | | **Test Designed Date: 16-03-2023** | | |
| **Module Name**: View the Admin Page | | | **Test Executed By: Ms. Merin Manoj** | | |
| **Test Title: View the Admin all Pages** | | | **Test Execution Date: 16-03-2023** | | |
| **Description: Verify the credentials and the admin pages** | | |  | | |
| **Pre-Condition: User** would enter unique username | | | | | |
| **Step** | **Test Step** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/**  **Fai l)** |
| 1 | Navigate to the login page |  | Display login page |  |  |
| 2 | Valid username entered | nurturenest@gmail.com | Admin should successfully login | Admin successfully login | Pass |
| 3 | Valid password entered | Admin@123 |
| 4 | Click on View Vaccine |  | Admin should navigate to the page where the vaccines listed | Admin navigate to the page where the added vaccine listed | Pass |
| 5 | Click on Vaccine Request |  | View the vaccine request from the health centers | View the vaccine request from the health centers | Pass |
| 6 | Click on View Feeding Chart |  | Admin should navigate and view the feeding chart page | Admin navigate and view the feeding chart page | Pass |
| **Post-Condition: Credentials verified and navigates to different pages** | | | | | |

**Test Case 3:**

**Code**

import pytest

import time

import json

from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.common.action\_chains import ActionChains

from selenium.webdriver.support import expected\_conditions

from selenium.webdriver.support.wait import WebDriverWait

from selenium.webdriver.common.keys import Keys

from selenium.webdriver.common.desired\_capabilities import DesiredCapabilities

class TestApprove():

def setup\_method(self, method):

self.driver = webdriver.Chrome()

self.vars = {}

def teardown\_method(self, method):

self.driver.quit()

def test\_approve(self):

self.driver.get("http://127.0.0.1:8000/login/")

self.driver.set\_window\_size(1366, 728)

self.driver.find\_element(By.ID, "email").click()

self.driver.find\_element(By.ID, "email").send\_keys("nurturenest@gmail.com")

self.driver.find\_element(By.ID, "password").click()

self.driver.find\_element(By.ID, "password").send\_keys("Admin@123")

self.driver.find\_element(By.ID, "password").send\_keys(Keys.ENTER)

self.driver.find\_element(By.LINK\_TEXT, "Total Registered Parents").click()

self.driver.find\_element(By.CSS\_SELECTOR, ".activate").click()

self.driver.find\_element(By.CSS\_SELECTOR,"tr:nth-child(5).statusbutton").click()

self.driver.find\_element(By.CSS\_SELECTOR, ".activate").click()

self.driver.find\_element(By.CSS\_SELECTOR,"tr:nth-child(5).statusbutton").click()

self.driver.find\_element(By.LINK\_TEXT, "NurtureNest").click()

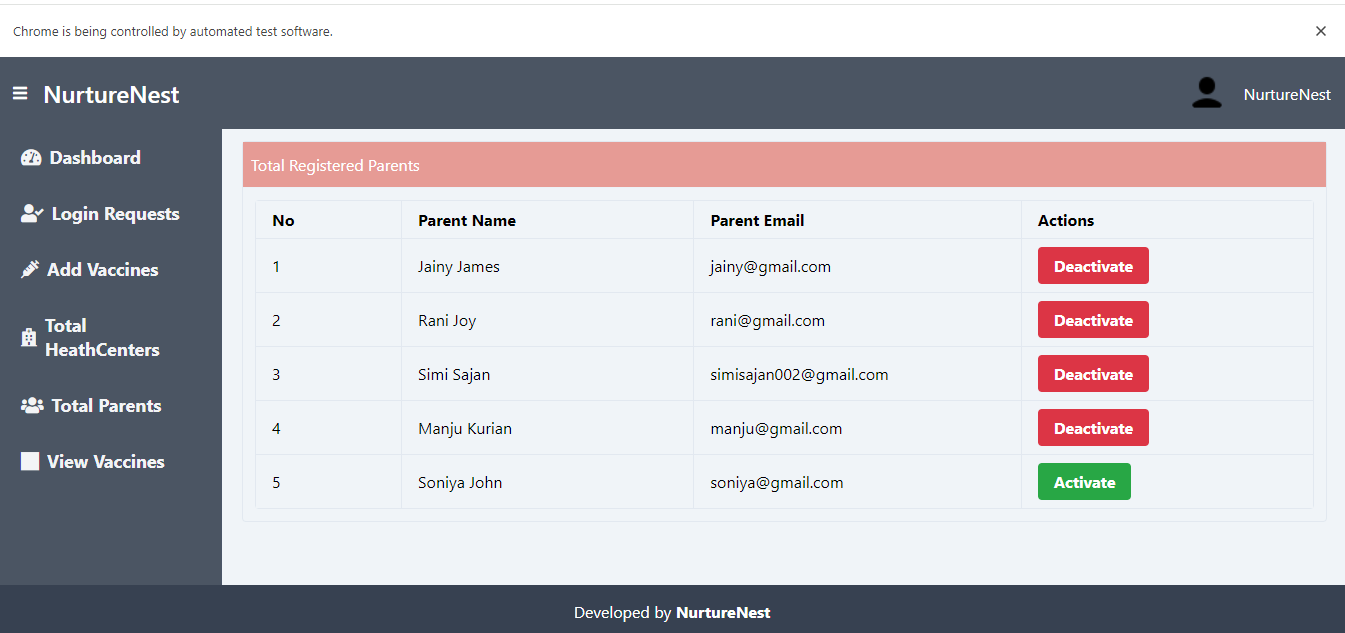
self.driver.find\_element(By.LINK\_TEXT, "Logout").click()

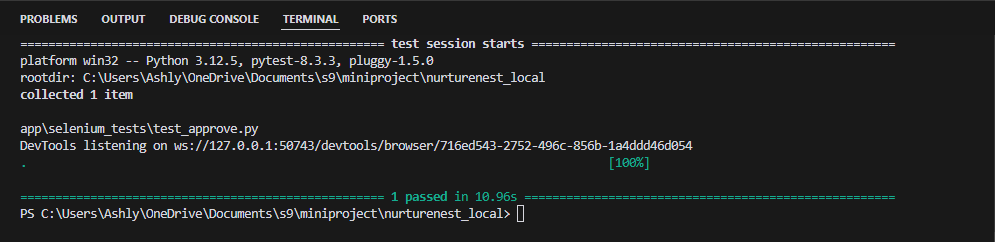
self.driver.close()

if \_\_name\_\_ == "\_\_main\_\_":

pytest.main([\_\_file\_\_])

**Screenshot**

****

****

**Test report**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case 3** | | | | | |
| **Project Name: NurtureNest** | | | | | |
| **Test Case to deactivate the User** | | | | | |
| **Test Case ID: Test\_3** | | | **Test Designed By: Ashlymol Thomas** | | |
| **Test Priority (Low/Medium/High): High** | | | **Test Designed Date: 16-03-2023** | | |
| **Module Name**: Deactivate User | | | **Test Executed By: Ms. Merin Manoj** | | |
| **Test Title: Deactivate and Activate the User** | | | **Test Execution Date: 16-03-2023** | | |
| **Description: Admin deactivate the user** | | |  | | |
| **Pre-Condition: User** would enter unique username | | | | | |
| **Step** | **Test Step** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/**  **Fai l)** |
| 1 | Navigate to the login page |  | Display login page |  |  |
| 2 | Valid username entered | nurturenest@gmail.com | Admin should successfully login to Admin home page | Admin successfully login to Admin home page | Pass |
| 3 | Valid password entered | Admin@123 |
| 4 | Click on Total Registered Parents | Click on the Deactivate Button | That particular user status should deactivate and not able to login | That particular user status gets deactivated and not able to login | Pass |
| **Post-Condition: Admin Deactivate the user and that particular user cannot able to login to their home page** | | | | | |

**Test Case 4:**

**Code**

import pytest

import time

from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.common.keys import Keys

class TestInvalidData():

def setup\_method(self, method):

self.driver = webdriver.Chrome()

self.vars = {}

def teardown\_method(self, method):

self.driver.quit()

def test\_invalidData(self):

self.driver.get("http://127.0.0.1:8000/login/")

self.driver.set\_window\_size(1382, 744)

self.driver.find\_element(By.ID, "email").click()

self.driver.find\_element(By.ID, "email").send\_keys("nurturenest@gmail.com")

self.driver.find\_element(By.ID, "password").click()

self.driver.find\_element(By.ID, "password").send\_keys("Admin@123")

self.driver.find\_element(By.ID, "password").send\_keys(Keys.ENTER)

self.driver.find\_element(By.LINK\_TEXT, "Add Vaccines").click()

self.driver.find\_element(By.ID, "b\_no").click()

self.driver.find\_element(By.ID, "b\_no").send\_keys("tyyy")

self.driver.find\_element(By.CSS\_SELECTOR, ".main-content").click()

self.driver.find\_element(By.ID, "b\_no").clear()

self.driver.find\_element(By.ID, "b\_no").send\_keys("o-00-====")

try:

error\_message = self.driver.find\_element(By.CSS\_SELECTOR, ".error-message

print("Error message displayed:", error\_message.text)

except:

print("No error message found.")

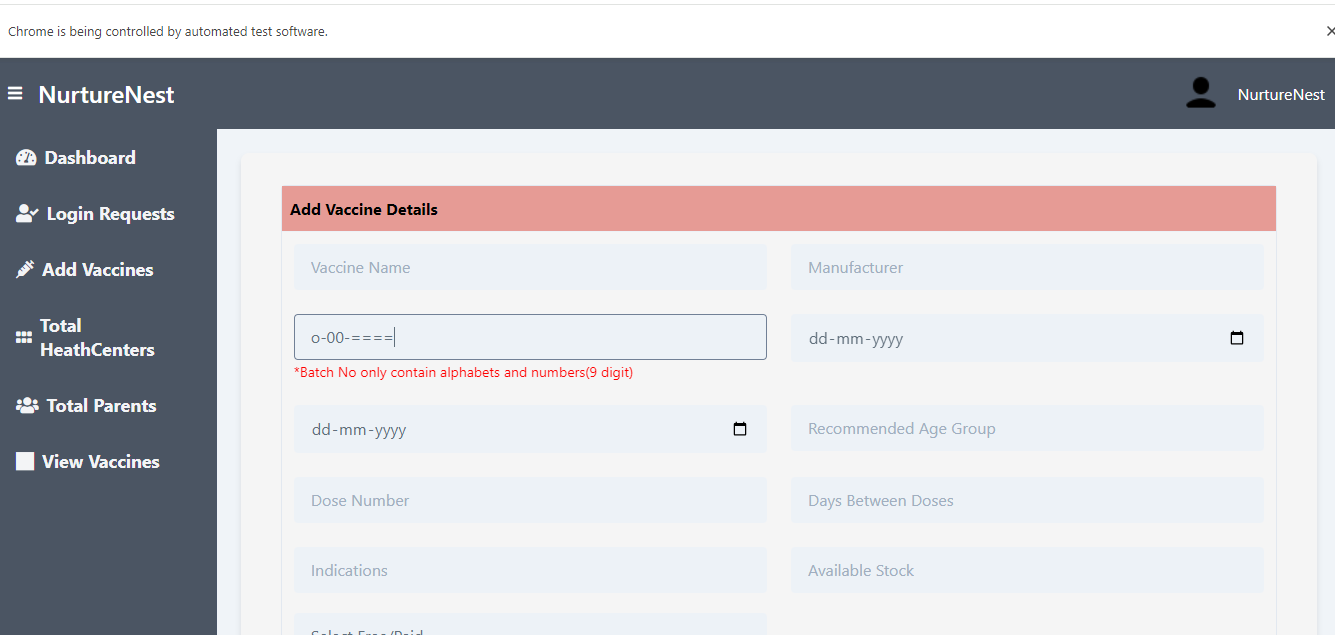
time.sleep(5) # Adjust time as needed to view error message

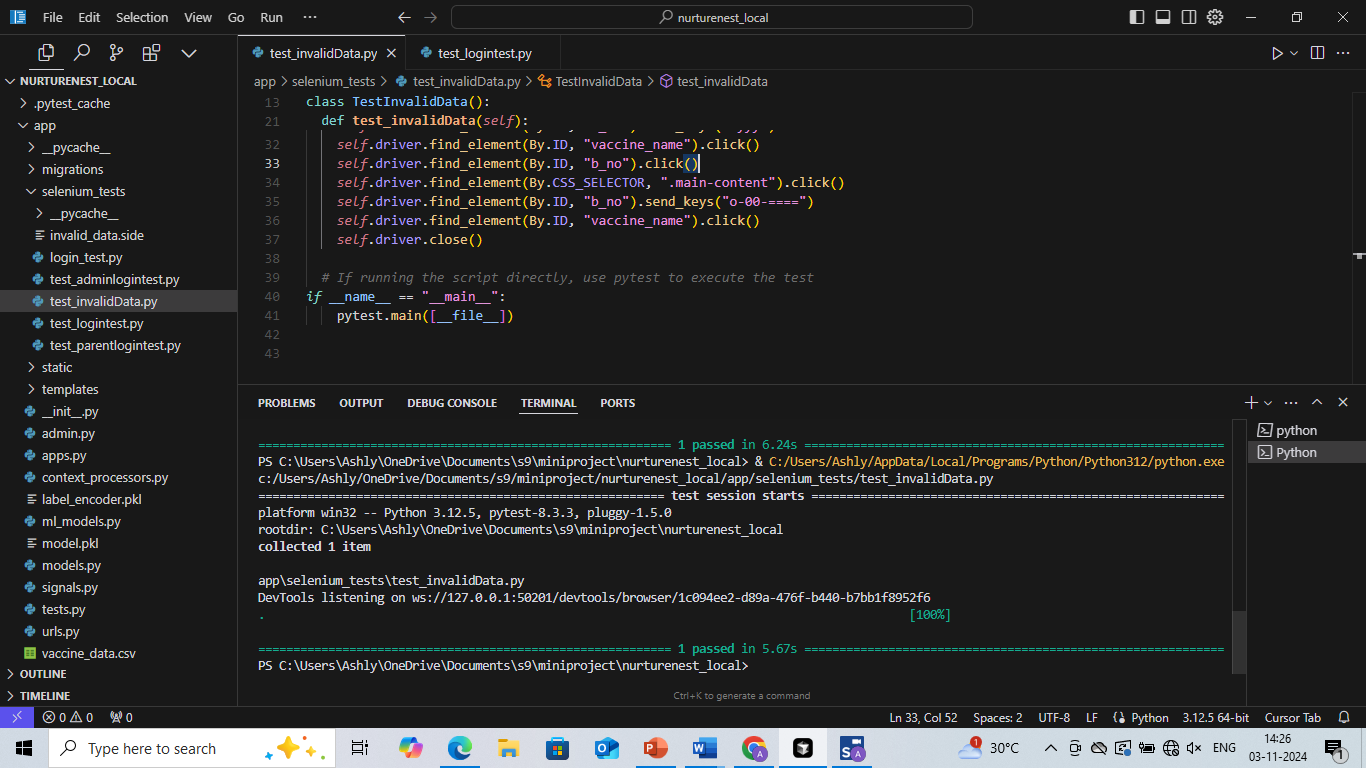
self.driver.close()

if \_\_name\_\_ == "\_\_main\_\_":

pytest.main([\_\_file\_\_])

**Screenshot**





**Test report**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case 4** | | | | | |
| **Project Name: NurtureNest** | | | | | |
| **User Registration Test Case** | | | | | |
| **Test Case ID: Test\_3** | | | **Test Designed By: Ashlymol Thomas** | | |
| **Test Priority (Low/Medium/High): High** | | | **Test Designed Date: 16-03-2023** | | |
| **Module Name**: Add Vaccines | | | **Test Executed By: Ms. Merin Manoj** | | |
| **Test Title: Add Invalid Vaccine Details** | | | **Test Execution Date: 16-03-2023** | | |
| **Description: Adding Invalid Vaccine Details** | | |  | | |
| **Pre-Condition: User** would enter unique username | | | | | |
| **Step** | **Test Step** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/**  **Fai l)** |
| 1 | Navigate to the login page |  | Display login page |  |  |
| 2 | Valid username entered | nurturenest@gmail.com | Admin should successfully login to Admin home page | Admin successfully login to Admin home page | Pass |
| 3 | Valid password entered | Admin@123 |
| 4 | Click on Add vaccine | Type Invalid Batch Number  00---00 | Must show the error message | show the error message | Pass |
| **Post-Condition: If the invalid format is added to vaccine details an error message will display** | | | | | |

# CHAPTER 6

# IMPLEMENTATION

## INTRODUCTION

In this section, we delve into the development and purpose of the Vaccine Image Classification and Details Retrieval System. This machine learning-driven system is designed to streamline the process of identifying various vaccinations by analyzing images and retrieving essential information related to each vaccine. Its primary function is to assist healthcare providers, parents, and medical personnel in quickly and accurately identifying vaccinations, thereby providing immediate access to critical information such as age suitability, intended purpose, and potential side effects. This automated solution not only enhances operational efficiency in medical environments but also improves data accessibility for users, ensuring that crucial details about vaccines are readily available at the point of need. By leveraging machine learning algorithms and structured datasets, the system achieves accurate classification of vaccine images, which can facilitate better-informed decisions and reduce human error.

## IMPLEMENTATION PROCEDURES

The implementation of the Vaccine Image Classification and Details Retrieval System involves a series of carefully planned steps that cover data preparation, model training, and system integration. Initially, vaccine images are gathered and preprocessed to ensure consistency, a critical step for model accuracy. Using grayscale conversion and resizing, images are standardized and transformed into feature vectors, making them suitable for classification tasks. Once preprocessed, the data is used to train a Support Vector Classifier (SVC), which associates image features with vaccine classifications. Post-training, the model is deployed, allowing users to upload images for real-time vaccine identification and information retrieval. The structured dataset, containing details like age range, purpose, and side effects, supports this system by providing necessary metadata for each classified vaccine. This section highlights each technical aspect of implementation to clarify the procedures undertaken to establish a functional and reliable system.

### User Training

User training is a critical component in ensuring effective system adoption and user satisfaction. In this phase, medical personnel and administrative staff are introduced to the system's functionalities, including image uploading, result interpretation, and data retrieval. Users are trained on how to upload clear, quality images to maximize classification accuracy and on how to interpret the retrieved information effectively. Tutorials, guided sessions, and practice scenarios are utilized to familiarize users with the system interface and workflow. Special emphasis is placed on understanding the system’s output and the implications of any side-effect warnings associated with certain vaccines. Overall, this training empowers users to confidently operate the system, ensuring its smooth integration into daily medical operations.

### Training on the Application Software

Training on the application software includes both technical instruction and practice sessions to enhance user proficiency with the system. This involves hands-on guidance on using the software interface, from initial login to navigating through menus and accessing various functions. The goal of this training is to make users comfortable with performing classification tasks, viewing vaccine details, and managing image files within the system. Users also learn how to interpret the model's confidence scores and manage any classification discrepancies. Additionally, troubleshooting sessions help users resolve common issues independently, fostering a sense of self-sufficiency and reducing reliance on technical support for routine tasks.

### System Maintenance

To ensure ongoing accuracy and reliability, system maintenance is essential. Maintenance procedures include routine checks on the machine learning model's performance, dataset updates to incorporate new vaccines, and software patches to address any emerging security or functionality issues. Regular re-training of the classification model may be necessary as new vaccine images are added to the dataset, ensuring that the system remains up-to-date with the latest data. Technical support staff are trained to carry out system updates and troubleshoot user-reported issues. Scheduled maintenance periods allow for thorough checks, and real-time monitoring ensures swift response to any operational issues, thus guaranteeing consistent and reliable performance for end-users.

* + 1. **Hosting**

Hosting a Python Django project involves deploying the application to a server so that users can access it over the internet. This process typically requires setting up a web server, configuring a domain, and ensuring the server environment supports Django and any dependencies. A common hosting approach is to use cloud providers, such as Amazon Web Services (AWS), Google Cloud Platform (GCP), or Microsoft Azure, which offer scalable

infrastructure and a variety of hosting options tailored for web applications. Another popular

choice is deploying on Platform-as-a-Service (PaaS) options like Heroku or PythonAnywhere,

which simplify the deployment process by handling much of the infrastructure setup.

For Django applications, a crucial step in deployment is setting up a reliable web server gateway interface (WSGI) or asynchronous gateway interface (ASGI) server, such as Gunicorn or Daphne, to interface between Django and the web server. Additionally, a reverse proxy server like Nginx or Apache is often used to handle client requests, manage static files, and improve application performance and security. Database configuration is also essential, as Django applications in production often require more robust databases like PostgreSQL or MySQL instead of the default SQLite, which is primarily intended for development.

Security is another key aspect in hosting Django applications. It involves setting up HTTPS with SSL certificates, securing sensitive data in environment variables, and configuring Django settings to limit potential security risks. To ensure ongoing reliability, developers often implement automated deployment processes, error logging, and monitoring tools, which provide real-time alerts and analytics on application health. Finally, it is essential to conduct thorough testing before and after deployment to identify any issues and to optimize the application for performance under expected traffic loads.

**Eg.Render**

Hosting a Django project on Render is an efficient and straightforward solution for deploying web applications. Render provides a fully managed platform that automates much of the deployment and scaling process, making it an ideal choice for Django applications. To host the project on Render, the process begins by connecting the GitHub or GitLab repository to Render, enabling continuous deployment whenever new code is pushed. Render’s integration with Git simplifies version control and updates, ensuring that the latest version of the Django application is always live without requiring manual intervention.

Render provides a user-friendly interface for setting up the necessary environment variables directly in the dashboard, allowing sensitive information such as secret keys and database credentials to be securely managed. When deploying a Django project, it’s also essential to set up a PostgreSQL database, as Render offers managed databases with easy configuration options. With a managed database, Render automatically handles database backups and scaling, reducing the administrative burden on developers.

The deployment process on Render involves configuring a render.yaml or using the Render dashboard to specify build and runtime settings, such as installing necessary dependencies, running migrations, and collecting static files. Additionally, Render’s HTTPS support ensures that applications are served securely over SSL, enhancing security without additional configuration. For efficient performance, Render allows scaling configurations, so applications can automatically adjust to handle higher traffic loads if needed. Overall, hosting a Django project on Render provides a seamless experience with tools for scaling, secure environment management, and continuous deployment, making it a robust choice for production-ready Django applications.

**Procedure for hosting a website on 000Webhost:**

Step 1: Ensure your Django project is in a Git repository on platforms like GitHub, GitLab, or Bitbucket. If not already done, initialize Git in your project folder, add all files with git add., and commit with git commit -m "Initial commit". Push the project to your remote repository.

Step 2: Go to Render.com and sign up for an account. Once logged in, click on **New +** > **Web Service** and connect your GitHub, GitLab, or Bitbucket account to select your project repository.

Step 3: Configure the web service settings by providing a name, selecting a region, and choosing the branch to deploy. In the **Build Command**, enter pip install -r requirements.txt, and for the **Start Command**, use gunicorn project\_name.wsgi --preload --bind 0.0.0.0:$PORT, replacing project\_name with your actual Django project name.

Step 4: Set up environment variables in the **Environment** section, including your SECRET\_KEY and setting DEBUG to False. Configure database connection settings if applicable.

Step 5: Click **Create Web Service** to deploy your Django project. Monitor the deployment process in the dashboard. Once completed, Render will provide a URL for your live application.

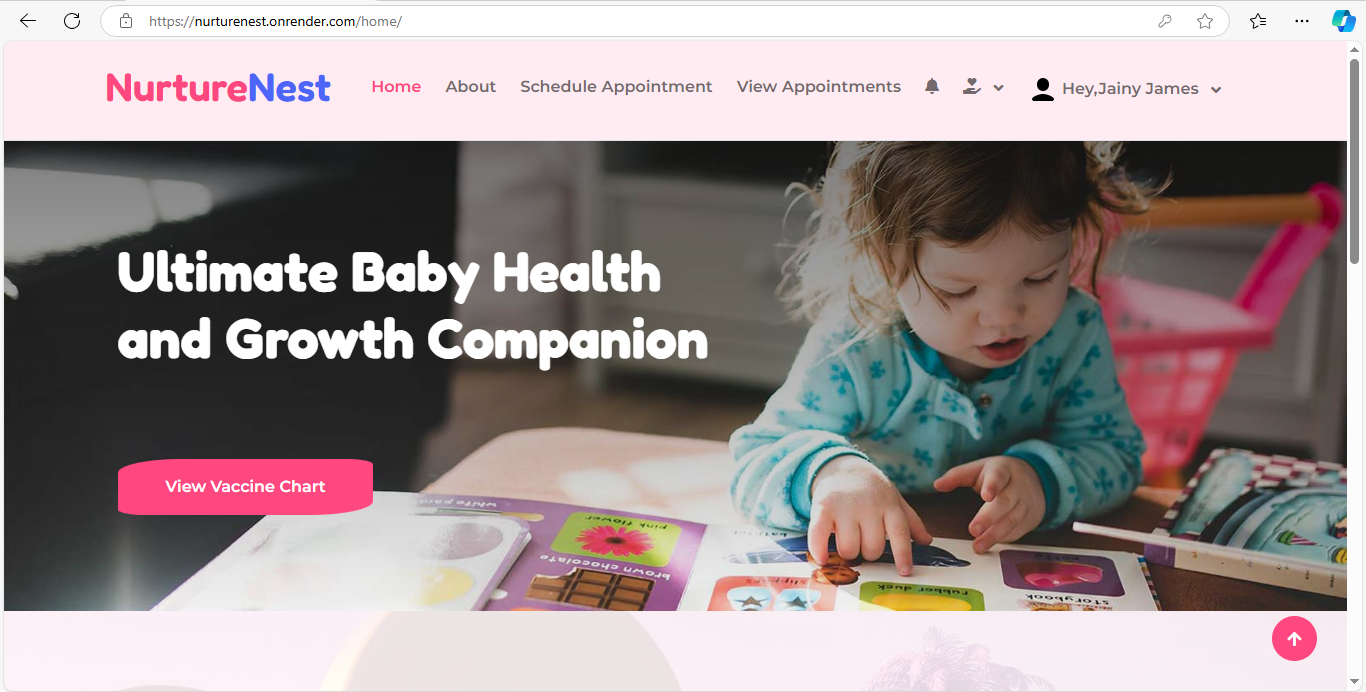
Step 6: Test your application at the provided URL. If there are issues, check the logs for debugging. You can update your code in the Git repository, and Render will automatically redeploy the changes.

**Hosted Website: render**

**Hosted Link:** <https://nurturenest.onrender.com>

**Hosted Link QR Code**

**Screenshot**



# CHAPTER 7

# CONCLUSION AND FUTURE SCOPE

## CONCLUSION

NurtureNest stands as an innovative and comprehensive solution for managing child health and development, seamlessly integrating various functionalities that cater to the needs of parents, healthcare providers, and administrators. By focusing on critical aspects such as vaccination scheduling, feeding guidance, and mental health support, the platform empowers users to take proactive steps in ensuring the well-being of children. The incorporation of machine learning enhances efficiency in healthcare workflows, automating tasks such as vaccine documentation and inventory management, which ultimately leads to better health outcomes. With its user-friendly interface and robust features, NurtureNest not only facilitates improved communication between parents and healthcare professionals but also fosters a community dedicated to nurturing children's health.

* 1. **FUTURE SCOPE**

The future development of NurtureNest holds immense potential for enhancing its capabilities and broadening its impact. Expanding the platform to include telemedicine features would enable virtual consultations between parents and healthcare providers, ensuring that children receive timely medical advice without the need for in-person visits. Integrating data analytics can provide parents and providers with deeper insights into vaccination trends, nutritional needs, and developmental milestones, facilitating more informed decision-making. Additionally, incorporating multilingual support could make NurtureNest accessible to a wider audience, promoting inclusivity and diversity in child health management. As technology continues to evolve, exploring partnerships with health organizations and educational institutions can further enhance the platform’s offerings, ultimately transforming NurtureNest into a leading resource for child health and development on a global scale.

# CHAPTER 8

# BIBLIOGRAPHY

### REFERENCES:

* + - ..
    - ..
    - ..
    - ..
    - ...

### WEBSITES:

* + - [..](http://www.w3schools.com/)
    - [..](http://www.jquery.com/)
    - [..](http://homepages.dcc.ufmg.br/%7Erodolfo/es-1-03/IEEE-Std-830-1998.pdf)
    - [..](http://www.agilemodeling.com/artifacts/useCaseDiagram.html)

# CHAPTER 9

# APPENDIX

## Sample Code

**Adding Vaccine Details**

Add\_vaccine.html

{% load static %}

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<meta name="keywords" content="tailwind,tailwindcss,tailwind css,css,starter template,free template,admin templates, admin template, admin dashboard, free tailwind templates, tailwind example">

<!-- Css -->

<link rel="stylesheet" href="{% static 'dist/styles.css' %}">

<link rel="stylesheet" href="{% static 'dist/all.css' %}">

<link href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:400,400i,600,600i,700,700i" rel="stylesheet">

<title>Dashboard | Admin</title>

<style>

.bg-nav {

background-color: #4B5563; /\* Dark gray for nav background \*/

}

body {

background-color: #f0f4f8; /\* Light gray background \*/

}

.no-underline {

text-decoration: none;

}

.text-black {

color: black;

}

.font-bold {

font-weight: bold;

}

.sidebar-link {

display: flex;

align-items: center;

padding: 1rem;

color: #F3F4F6; /\* Light gray text \*/

font-weight: bold;

font-size: 18px;

}

.sidebar-link:hover {

background-color: #1F2937; /\* Darker gray on hover \*/

}

.main-content {

padding: 2rem;

background-color: #f5f5f5; /\* White main content area \*/

box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1); /\* Subtle shadow for depth \*/

border-radius: 8px; /\* Rounded corners \*/

}

.footer {

background-color: #374151; /\* Darker gray footer \*/

color: white;

padding: 1rem;

text-align: center;

}

.icon-style {

font-size: 2rem; /\* Larger icons \*/

}

.box-title {

margin-top: 1rem;

font-size: 1.25rem; /\* Larger text \*/

}

.box-container {

display: flex;

flex-wrap: wrap;

justify-content: space-between;

}

.box-container > div {

flex: 1 1 calc(33.333% - 20px); /\* Three boxes per row with space between \*/

margin-bottom: 20px; /\* Space between rows \*/

}

@media (max-width: 768px) {

.box-container > div {

flex: 1 1 calc(100% - 20px); /\* Single box per row on smaller screens \*/

}

}

.profile-container {

position: relative; /\* Positioning for dropdown \*/

display: flex; /\* Align items horizontally \*/

align-items: center; /\* Center vertically \*/

}

.profile-dropdown {

background-color: #4B5563; /\* Dark gray background \*/

color: white; /\* White text \*/

padding: 10px; /\* Padding \*/

text-align: left; /\* Align text left \*/

width: 200px; /\* Set width \*/

box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1); /\* Subtle shadow \*/

border-radius: 8px; /\* Rounded corners \*/

position: absolute; /\* Position dropdown absolutely \*/

top: 100%; /\* Position below the container \*/

right: 0; /\* Align to the right of the container \*/

margin-top: 8px; /\* Space between text and dropdown \*/

display: none; /\* Initially hidden \*/

z-index: 10; /\* Ensure dropdown is above other content \*/

}

.profile-dropdown a {

color: white; /\* White text \*/

padding: 10px;

display: block;

text-decoration: none; /\* No underline \*/

}

.profile-dropdown a:hover {

background-color: #1F2937; /\* Darker gray on hover \*/

}

.show {

display: block; /\* Display when shown \*/

}

.error {

color: red;

font-size: 0.875rem;

}

.text-success {

color: green;

font-size: 0.875rem;

}

</style>

</head>

<body>

<!--Container -->

<div class="mx-auto bg-grey-400">

<!--Screen-->

<div class="min-h-screen flex flex-col">

<!--Header Section Starts Here-->

<header class="bg-nav p-4">

<div class="flex justify-between">

<div class="inline-flex items-center">

<i class="fas fa-bars pr-4 text-white" onclick="sidebarToggle()"></i>

<h1 class="text-white text-2xl">NurtureNest</h1>

</div>

<div class="flex items-center profile-container">

<img onclick="profileToggle()" class="h-10 w-10 rounded-full" src="{% static 'img/name.png' %}" alt="Profile">

<a href="#" onclick="profileToggle()" class="text-white ml-4 hidden md:block lg:block">NurtureNest</a>

<div id="ProfileDropDown" class="profile-dropdown hidden shadow-md absolute mt-12 mr-4 right-0">

<a href="#">My Profile</a>

<a href="{% url 'logout' %}">Logout</a>

</div>

</div>

</div>

</header>

<!--/Header-->

<div class="flex flex-1">

<!--Sidebar-->

<aside id="sidebar" class="bg-nav w-1/5 md:w-1/6 lg:w-1/6 border-r border-gray-700 hidden md:block lg:block">

<ul class="list-reset">

<li class="sidebar-link">

<i class="fas fa-tachometer-alt mx-2"></i>

<a href="{% url 'admin\_home' %}" class="no-underline">Dashboard</a>

</li>

<li class="sidebar-link">

<i class="fas fa-user-check mx-2"></i>

<a href="{% url 'request' %}" class="no-underline">Login Requests</a>

</li>

<li class="sidebar-link">

<i class="fas fa-syringe mx-2"></i>

<a href="{% url 'add\_vaccine' %}" class="no-underline">Add Vaccines</a>

</li>

<li class="sidebar-link">

<i class="fas fa-grip-horizontal mx-2"></i>

<a href="{% url 'total\_healthcenters' %}" class="no-underline">Total HeathCenters</a>

</li>

<li class="sidebar-link">

<i class="fas fa-users mx-2"></i>

<a href="{% url 'total\_parents' %}" class="no-underline">Total Parents</a>

</li>

<li class="sidebar-link">

<i class="fas fa-square-full mx-2"></i>

<a href="{% url 'view\_vaccines' %}" class="no-underline">View Vaccines</a>

</li>

</ul>

</aside>

<!--/Sidebar-->

<!--Main-->

<main class="flex-1 p-6">

<div class="main-content">

<!--Grid Form-->

<body>

<div class="flex flex-1 flex-col md:flex-row lg:flex-row mx-2">

<div class="mb-2 border-solid border-gray-300 rounded border shadow-sm w-full">

<div style="background-color: #e69b95;" class="bg-gray-200 px-2 py-3 border-solid border-gray-200 border-b">

<b>Add Vaccine Details</b>

</div>

<div class="p-3">

{% if messages %}

<div>

{% for message in messages %}

<div class="bg-green-300 mb-2 border border-green-300 text-green-600 px-4 py-3 rounded relative" role="alert">

<strong class="font-bold">{{ message.tags|title }}!</strong>

<span class="block sm:inline">{{ message }}</span>

<span class="absolute top-0 bottom-0 right-0 px-4 py-3">

<svg class="fill-current h-6 w-6 text-green" role="button" xmlns="http://www.w3.org/2000/svg" viewBox="0 0 20 20"><title>Close</title><path d="M14.348 14.849a1.2 1.2 0 0 1-1.697 0L10 11.819l-2.651 3.029a1.2 1.2 0 1 1-1.697-1.697l2.758-3.15-2.759-3.152a1.2 1.2 0 1 1 1.697-1.697L10 8.183l2.651-3.031a1.2 1.2 0 1 1 1.697 1.697l-2.758 3.152 2.758 3.15a1.2 1.2 0 0 1 0 1.698z"/></svg>

</span>

</div>

{% endfor %}

</div>

{% endif %}

<form class="w-full" method="post" action="{% url 'add\_vaccine' %}">

{% csrf\_token %}

<div class="flex flex-wrap -mx-3 mb-6">

<div class="w-full md:w-1/2 px-3 mb-6 md:mb-0">

<input class="appearance-none block w-full bg-gray-200 text-grey-darker border border-gray-200 rounded py-3 px-4 leading-tight focus:outline-none focus:bg-white-500 focus:border-gray-600"

id="vaccine\_name" name="vaccine\_name" type="text" placeholder="Vaccine Name">

<span class="error" id="nameError" style="display:none;">\*Vaccine name should contain only alphabets, numbers, and hyphens</span>

</div>

<div class="w-full md:w-1/2 px-3">

<input class="appearance-none block w-full bg-gray-200 text-grey-darker border border-gray-200 rounded py-3 px-4 leading-tight focus:outline-none focus:bg-white-500 focus:border-gray-600"

id="m\_name" name="manufacturer" type="text" placeholder="Manufacturer">

<span class="error" id="mError" style="display:none;">\*It should contain only alphabets</span>

</div>

</div>

<div class="flex flex-wrap -mx-3 mb-6">

<div class="w-full md: w-1/2 px-3 mb-6 md: mb-0">

<input class="appearance-none block w-full bg-gray-200 text-grey-darker border border-gray-200 rounded py-3 px-4 leading-tight focus:outline-none focus:bg-white-500 focus:border-gray-600"

id="b\_no" name="batch\_number" type="text" placeholder="Batch Number">

<span class="error" id="bError" style="display:none;">\*Batch No only contain alphabets and numbers(9 digit)</span>

</div>

<div class="w-full md:w-1/2 px-3">

<input class="appearance-none block w-full bg-gray-200 text-grey-darker border border-gray-200 rounded py-3 px-4 leading-tight focus:outline-none focus:bg-white-500 focus:border-gray-600"

id="date\_mnf" name="date\_manufacture" type="date" placeholder="Date of Manufacture" required>

</div>

</div>

<div class="flex flex-wrap -mx-3 mb-6">

<div class="w-full md:w-1/2 px-3">

<input class="appearance-none block w-full bg-gray-200 text-grey-darker border border-gray-200 rounded py-3 px-4 leading-tight focus:outline-none focus:bg-white-500 focus:border-gray-600"

id="exp\_date" name="expiry\_date" type="date" placeholder="Expiry Date" required min="{{ today }}">

</div>

<div class="w-full md:w-1/2 px-3">

<input class="appearance-none block w-full bg-gray-200 text-grey-darker border border-gray-200 rounded py-3 px-4 leading-tight focus:outline-none focus:bg-white-500 focus:border-gray-600"

id="age" name="age\_group" type="text" placeholder="Recommended Age Group">

<span class="error" id="ageError" style="display:none;">\*Enter Valid Age</span>

</div>

</div>

<div class="flex flex-wrap -mx-3 mb-6">

<div class="w-full md:w-1/2 px-3 mb-6 md:mb-0">

<input class="appearance-none block w-full bg-gray-200 text-grey-darker border border-gray-200 rounded py-3 px-4 leading-tight focus:outline-none focus:bg-white-500 focus:border-gray-600"

id="dose" name="dose\_number" type="number" placeholder="Dose Number" required min="0" max="5">

<span class="error" id="doseError" style="display:none;">\*Enter a valid dose (0 to 5)</span>

</div>

<div class="w-full md:w-1/2 px-3">

<input class="appearance-none block w-full bg-gray-200 text-grey-darker border border-gray-200 rounded py-3 px-4 leading-tight focus:outline-none focus:bg-white-500 focus:border-gray-600"

id="interval\_days" name="interval\_days" type="text" placeholder="Days Between Doses">

<span class="error" id="ageError" style="display:none;">\*Enter Valid Age</span>

</div>

</div>

<div class="flex flex-wrap -mx-3 mb-5">

<div class="w-full md:w-1/2 px-3 mb-6 md:mb-0">

<input class="appearance-none block w-full bg-gray-200 text-grey-darker border border-gray-200 rounded py-3 px-4 leading-tight focus:outline-none focus:bg-white-500 focus:border-gray-600"

id="indications" name="indications" type="text" placeholder="Indications">

<span class="error" id="indicationError" style="display:none;">\*Enter Valid Indications</span>

</div>

<div class="w-full md:w-1/2 px-3 mb-6 md:mb-0">

<input class="appearance-none block w-full bg-gray-200 text-grey-darker border border-gray-200 rounded py-3 px-4 leading-tight focus:outline-none focus:bg-white-500 focus:border-gray-600"

id="stock" name="stock" type="number" placeholder="Available Stock" required min="0" max="100">

<span class="error" id="stockError" style="display:none;">\*Enter valid stock (0 to 100)</span>

</div>

</div>

<div class="flex flex-wrap -mx-3 mb-6">

<div class="w-full md:w-1/2 px-3">

<select id="free-paid" name="free\_or\_paid" onchange="toggleRateField()" class="appearance-none block w-full bg-gray-200 text-grey-darker border border-gray-200 rounded py-3 px-4 leading-tight focus:outline-none focus:bg-white-500 focus:border-gray-600">

<option value="" selected disabled>Select Free/Paid</option>

<option value="free">Free</option>

<option value="paid">Paid</option>

</select>

<span class="error" id="paymentError" style="display:none;">\*Enter Free/Paid</span>

</div>

<div class="flex flex-wrap -mx-3 mb-6" id="rate-field" style="display:none;">

<div class="w-full md:w-1/2 px-3">

<input class="appearance-none block w-full bg-gray-200 text-grey-darker border border-gray-200 rounded py-3 px-4 leading-tight focus:outline-none focus:bg-white-500 focus:border-gray-600"

id="rate" name="rate" type="number" placeholder="Rate">

</div>

</div></div>

<div class="flex flex-wrap -mx-3 mb-6">

<div class="w-full px-3">

<button style="background-color: #e69b95;" type="submit" class="appearance-none block w-full bg-gray-200 text-grey-darker border border-gray-200 rounded py-3 px-4 leading-tight focus:outline-none focus:bg-white-500 focus:border-gray-600">

Add Vaccine

</button>

</div>

</div>

</form>

</div>

</div></div>

<! --/Grid Form-->

</div>

</main>

<!--/Main-->

</div></div></div>

<script>

function sidebarToggle() {

document.getElementById('sidebar').classList.toggle('hidden');

}

function profileToggle() {

document.getElementById('ProfileDropDown').classList.toggle('hidden');

}

function toggleRateField() {

const freePaidSelect = document.getElementById('free-paid');

const rateField = document.getElementById('rate-field');

if (freePaidSelect.value === 'paid') {

rateField.style.display = 'block';

} else {

rateField.style.display = 'none';

}

}

document.getElementById('exp\_date').addEventListener('change', function() {

var expDate = this.value; // This will be in YYYY-MM-DD format

console.log('Selected date:', expDate);

});

</script><script>

document.addEventListener('DOMContentLoaded', function() {

// Get today's date in YYYY-MM-DD format

const today = new Date();

const dd = String(today.getDate()).padStart(2, '0');

const mm = String(today.getMonth() + 1).padStart(2, '0'); // January is 0!

const yyyy = today.getFullYear();

const formattedDate = yyyy + '-' + mm + '-' + dd;

document.getElementById('exp\_date').setAttribute('min', formattedDate);

});

</script>

<script>

document.addEventListener('DOMContentLoaded', function() {

const today = new Date();

const dd = String(today.getDate()).padStart(2, '0');

const mm = String(today.getMonth() + 1).padStart(2, '0'); // January is 0!

const yyyy = today.getFullYear();

const formattedDate = yyyy + '-' + mm + '-' + dd;

document.getElementById('exp\_date').setAttribute('min', formattedDate);

});

</script><script>

$(document).ready(function() {

function validatedose() {

var dose = $("#dose").val();

// Check if the dose is a number and within the range of 0 to 5

if (dose === "" || isNaN(dose) || dose < 0 || dose > 5) {

$("#doseError").show();

return false;

} else {

$("#doseError").hide();

return true;

}

}

});</script>

<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>

<script>

$(document).ready(function() {

$(".error").hide();

$("#vaccine\_name").keyup(validateVaccineName);

$("#m\_name").keyup(validateMName);

$("#b\_no").keyup(validateBno);

$("#age").keyup(validateAge);

$("#dose").keyup(validatedose);

$("#indications").keyup(validateIndications);

$("#stockError").keyup(validatestock);

function validateVaccineName() {

var vaccineNameRegex = /^[A-Za-z0-9\s-\/]+$/;

var vaccineName = $("#vaccine\_name").val().trim();

if (!vaccineNameRegex.test(vaccineName)) {

$("#nameError").show();

return false;

} else {

$("#nameError").hide();

return true;

}

}

function validateMName() {

var nameRegex = /^[A-Za-z\s-]+$/;

var manufacturerName = $("#m\_name").val();

if (!nameRegex.test(manufacturerName)) {

$("#mError").show();

return false;

} else {

$("#mError").hide();

return true;

}

}

function validateBno() {

// Regex to match batch numbers that:

// - Must be exactly 9 characters long

// - Can include alphabets (both uppercase and lowercase), numbers, and exactly one hyphen

var bnoRegex = /^[A-Za-z0-9]{4}[A-Za-z0-9]?-?[A-Za-z0-9]{4}$/;

var b\_no = $("#b\_no").val().trim();

if (!bnoRegex.test(b\_no) || b\_no.length !== 9) {

$("#bError").show(); // Show error if validation fails

return false;

} else {

$("#bError").hide(); // Hide error if validation passes

return true;

}

}

function validateAge() {

var ageInput = $("#age").val().trim();

var ageRegex = /^(?:([1-9]|[1-9]\d|[1-9]\d\d)(?:-(?:[1-9]|[1-9]\d|[1-9]\d\d))?)\s\*(months?|weeks?|years?)$/i;

var match = ageInput.match(ageRegex);

if (!match) {

$("#ageError").show();

return false;

}

var ageParts = ageInput.split(/\s+/);

var ageRange = ageParts[0].split('-');

if (ageRange.length === 2) {

var startAge = parseInt(ageRange[0], 10);

var endAge = parseInt(ageRange[1], 10);

if (startAge < 1 || endAge < 1 || startAge >= endAge || endAge > 999) {

$("#ageError").show();

return false;

}

} else {

var singleAge = parseInt(ageRange[0], 10);

if (singleAge < 1 || singleAge > 999) {

$("#ageError").show();

return false;

}

}

$("#ageError").hide();

return true;

}

function validatedose() {

var doseRegex = /^[1-5]$/;

var dose = $("#dose").val();

if (!doseRegex.test(dose)) {

$("#doseError").show();

return false;

} else {

$("#doseError").hide();

return true;

}

}

function validatestock() {

var stock = $("#stock").val();

// Check if the stock is a number and within the range of 0 to 100

if (stock === "" || isNaN(stock) || stock < 0 || stock > 100) {

$("#stockError").show();

return false;

} else {

$("#stockError").hide();

return true;

}}

function validateIndications() {

var indicationsRegex = /^[A-Za-z]+(?: [A-Za-z]+)\*$/;

var indications = $("#indications").val().trim();

if (!indicationsRegex.test(indications)) {

$("#iError").show(); // Show error if validation fails

return false;

} else {

$("#iError").hide(); // Hide error if validation passes

return true;

}}

$("#submit").click(function(event) {

event.preventDefault(); // Prevent form submission

var vaccine\_name = validateVaccineName();

var m\_name = validateMName();

var b\_no = validateBno();

var age = validateAge();

var dose = validatedose();

var indications = validateIndications();

var stock = validatestock();

if (vaccine\_name && m\_name && b\_no && age && dose && indications && stock) {

// Form is valid, proceed with form submission

$("form").submit();

}

});

});

</script>

<script type="text/javascript">

function preventBack() {

window.history.forward();

}

setTimeout("preventBack()", 0);

window.onunload = function () {

null

};

</script>

</body>

</html>

**View\_vaccines.html**

{% load static %}

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<!-- Css -->

<link rel="stylesheet" href="{% static 'dist/styles.css' %}">

<link rel="stylesheet" href="{% static 'dist/all.css' %}">

<link href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:400,400i,600,600i,700,700i" rel="stylesheet">

<title>Admin | Requests</title>

</head>

<style>

.bg-nav {

background-color: #4B5563;

}

body {

background-color: #f0f4f8;

}

.bg-box-color {

background-color: #FF8A80;

border-left-width: 6px;

height: 150px;

display: flex;

justify-content: center;

align-items: center;

margin: 10px;

border-left: 6px solid #FF6F61;

border-radius: 8px;

transition: background-color 0.3s, transform 0.3s;

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

}

.bg-box-color:hover {

background-color: #FF6F61;

transform: translateY(-5px);

}

.no-underline {

text-decoration: none;

}

.text-black {

color: black;

}

.font-bold {

font-weight: bold;

}

.sidebar-link {

display: flex;

align-items: center;

padding: 1rem;

color: #F3F4F6;

font-weight: bold;

font-size: 18px;

}

.sidebar-link:hover {

background-color: #1F2937;

}

.main-content {

padding: 2rem;

background-color: #FFFFFF;

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

border-radius: 8px;

}

.footer {

background-color: #374151;

color: white;

padding: 1rem;

text-align: center;

}

.icon-style {

font-size: 2rem;

}

.box-title {

margin-top: 1rem;

font-size: 1.25rem;

}

.box-container {

display: flex;

flex-wrap: wrap;

justify-content: space-between;

}

.box-container>div {

flex: 1 1 calc(33.333% - 20px);

margin-bottom: 20px;

}

@media (max-width: 768px) {

.box-container>div {

flex: 1 1 calc(100% - 20px);

}

}

.status-button {

padding: 0.5rem 1rem;

border-radius: 4px;

font-weight: bold;

color: white;

border: none;

cursor: pointer;

transition: background-color 0.3s;

}

.activate {

background-color: #28a745; /\* Green \*/

}

.deactivate {

background-color: #dc3545; /\* Red \*/

}

.status-button:hover {

opacity: 0.8;

}

.modal {

display: none;

position: fixed;

z-index: 1;

left: 0;

top: 0;

width: 100%;

height: 100%;

overflow: auto;

background-color: rgba(0, 0, 0, 0.4);

/\* Overlay color \*/

}

.modal-content {

background-color: white;

margin: 15% auto;

padding: 20px;

border: 1px solid #888;

width: 80%;

max-width: 400px;

text-align: center;

}

.modal-content p {

color: green;

font-weight: bold;

}

.close {

cursor: pointer;

}

.profile-container {

position: relative; /\* Positioning for dropdown \*/

display: flex; /\* Align items horizontally \*/

align-items: center; /\* Center vertically \*/

}

.profile-dropdown {

background-color: #4B5563; /\* Dark gray background \*/

color: white; /\* White text \*/

padding: 10px; /\* Padding \*/

text-align: left; /\* Align text left \*/

width: 200px; /\* Set width \*/

box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1); /\* Subtle shadow \*/

border-radius: 8px; /\* Rounded corners \*/

position: absolute; /\* Position dropdown absolutely \*/

top: 100%; /\* Position below the container \*/

right: 0; /\* Align to the right of the container \*/

margin-top: 8px; /\* Space between text and dropdown \*/

display: none; /\* Initially hidden \*/

z-index: 10; /\* Ensure dropdown is above other content \*/

}

.profile-dropdown a {

color: white; /\* White text \*/

padding: 10px;

display: block;

text-decoration: none; /\* No underline \*/

}

.profile-dropdown a:hover {

background-color: #1F2937; /\* Darker gray on hover \*/

}

.show {

display: block; /\* Display when shown \*/

}

.box-container {

display: flex;

flex-wrap: wrap;

justify-content: space-between;

}

.box-container > div {

flex: 1 1 calc(33.333% - 20px); /\* Three boxes per row with space between \*/

margin-bottom: 20px; /\* Space between rows \*/

}

@media (max-width: 768px) {

.box-container > div {

flex: 1 1 calc(100% - 20px); /\* Single box per row on smaller screens \*/

}

}

</style>

<body>

<!--Container -->

<div class="mx-auto bg-grey-400">

<!--Screen-->

<div class="min-h-screen flex flex-col">

<!--Header Section Starts Here-->

<header class="bg-nav p-4">

<div class="flex justify-between">

<div class="inline-flex items-center">

<i class="fas fa-bars pr-4 text-white" onclick="sidebarToggle()"></i>

<h1 class="text-white text-2xl">NurtureNest</h1>

</div>

<div class="flex items-center profile-container">

<img onclick="profileToggle()" class="h-10 w-10 rounded-full" src="{% static 'img/name.png' %}" alt="Profile">

<a href="#" onclick="profileToggle()" class="text-white ml-4 hidden md:block lg:block">NurtureNest</a>

<div id="ProfileDropDown" class="profile-dropdown hidden shadow-md absolute mt-12 mr-4 right-0">

<a href="{% url 'logout' %}">Logout</a>

</div>

</div>

</div>

</header>

<!--/Header-->

<div class="flex flex-1">

<!--Sidebar--> <aside id="sidebar" class="bg-nav w-1/5 md:w-1/6 lg:w-1/6 border-r border-gray-700 hidden md:block lg:block">

<ul class="list-reset">

<li class="sidebar-link">

<i class="fas fa-tachometer-alt mx-2"></i>

<a href="{% url 'admin\_home' %}" class="no-underline">Dashboard</a>

</li>

<li class="sidebar-link">

<i class="fas fa-user-check mx-2"></i>

<a href="{% url 'request' %}" class="no-underline">Login Requests</a>

</li>

<li class="sidebar-link">

<i class="fas fa-syringe mx-2"></i>

<a href="{% url 'add\_vaccine' %}" class="no-underline">Add Vaccines</a>

</li>

<li class="sidebar-link">

<i class="fas fa-grip-horizontal mx-2"></i>

<a href="{% url 'total\_healthcenters' %}" class="no-underline">Total HeathCenters</a>

</li>

<li class="sidebar-link">

<i class="fas fa-users mx-2"></i>

<a href="{% url 'total\_parents' %}" class="no-underline">Total Parents</a>

</li>

<li class="sidebar-link">

<i class="fas fa-square-full mx-2"></i>

<a href="{% url 'view\_vaccines' %}" class="no-underline">View Vaccines</a>

</li>

</ul>

</aside>

<main class="bg-white-500 flex-1 p-3 overflow-hidden">

<main class="flex-1 p-6">

<div class="main-content">

<!-- Stats Grid Starts Here -->

<div class="box-container">

<div class="flex flex-1 flex-col md:flex-row lg:flex-row mx-2">

<div class="mb-2 border-solid border-gray-300 rounded border shadow-sm w-full">

<div style="background-color: #e69b95; color: white;" class="px-2 py-3 border-solid border-gray-200 border-b">

<h2 class="text-xl font-semibold">Added Vaccines</h2>

</div>

<div class="p-3">

<table class="table-responsive w-full rounded">

<thead>

<tr>

<th class="border w-1/12 px-2 py-1 text-xs">No</th>

<th class="border w-1/6 px-2 py-1 text-xs">Vaccine Name</th>

<th class="border w-1/6 px-2 py-1 text-xs">Batch Number</th>

<th class="border w-1/6 px-2 py-1 text-xs">Age Group</th>

<th class="border w-1/6 px-2 py-1 text-xs">Stock</th>

<th class="border w-1/6 px-6 py-1 text-xs">Actions</th>

</tr>

</thead>

<tbody>

{% for vaccine in vaccines %}

<tr>

<td class="border px-2 py-1 text-xs">{{ forloop.counter }}</td>

<td class="border px-2 py-1 text-xs">{{ vaccine.vaccine\_name }}</td>

<td class="border px-2 py-1 text-xs">{{ vaccine.batch\_number }}</td>

<td class="border px-2 py-1 text-xs">{{ vaccine.age\_group }}</td>

<td class="border px-2 py-1 text-xs">{{ vaccine.stock }}</td>

<td class="border px-2 py-1 text-xs">

<div class="flex space-x-2">

<a href="{% url 'view\_vaccine\_details' vaccine.vaccine\_id %}">

<button type="submit" class="status-button activate"><i class="fas fa-eye"></i></button>

</a>

&nbsp;&nbsp;

<a href="#">

<button type="submit" class="status-button activate"><i class="fas fa-edit"></i></button>

</a>

&nbsp;&nbsp;

<a href="{% url 'delete\_vaccine' vaccine.vaccine\_id %}">

<button type="submit" class="status-button deactivate">X</button>

</a></div>

</td></tr>

{% empty %}

<tr>

<td colspan="6" class="text-center py-4">No vaccines available</td>

</tr>

{% endfor %}

</tbody>

</table>

</div></div></div></div>

<!--/Grid Form-->

</main>

<!--/Main-->

<br><br><br><br><br><br><br><br><br>

</div></div>

<!--Footer-->

<footer class="footer">

<p>Developed by <b>NurtureNest</b></p>

</footer>

<!--/Footer-->

<!-- Pop-up Modal -->

<div id="popupModal" class="modal">

<div class="modal-content">

<span class="close">&times;</span>

<p>Parent's status has been updated successfully.</p>

</div>

</div>

<!-- Scripts -->

<script>

// Toggle sidebar

function sidebarToggle() {

const sidebar = document.getElementById('sidebar');

sidebar.classList.toggle('hidden');

}

// Toggle profile dropdown

function profileToggle() {

const profileDropdown = document.getElementById('ProfileDropDown');

profileDropdown.classList.toggle('hidden');

profileDropdown.classList.toggle('show');

}

// Modal functionality

var modal = document.getElementById("popupModal");

var closeBtn = document.getElementsByClassName("close")[0];

closeBtn.onclick = function () {

modal.style.display = "none";

}

window.onclick = function (event) {

if (event.target == modal) {

modal.style.display = "none";

}

}

// Function to show the modal

function showModal() {

modal.style.display = "block";

}

</script>

<script type="text/javascript">

function preventBack() {

window.history.forward();

}

setTimeout("preventBack()", 0);

window.onunload = function () {

null

};

</script>

</body>

</html>

## Screen Shots

## 

## Fig:1 : Login Page

## 

## Fig:1 : Register Page

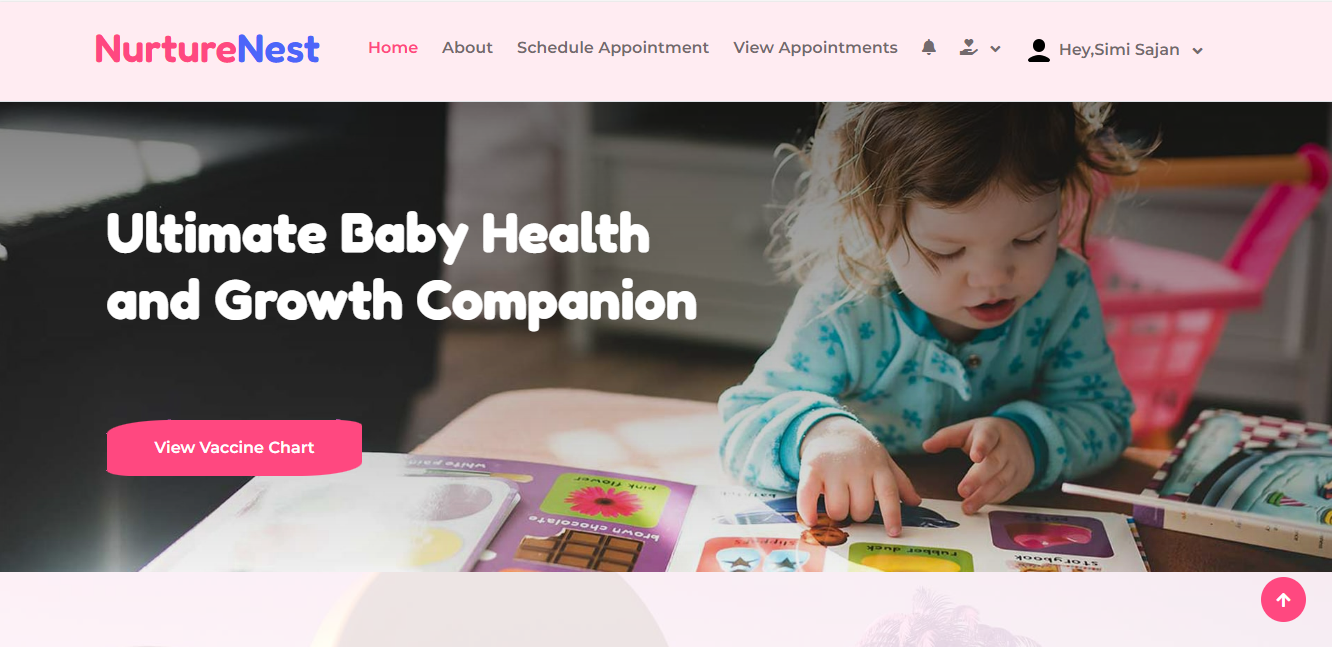
****

Fig:1 : Home Page

****

Fig:1 : Selecting Vaccine Page

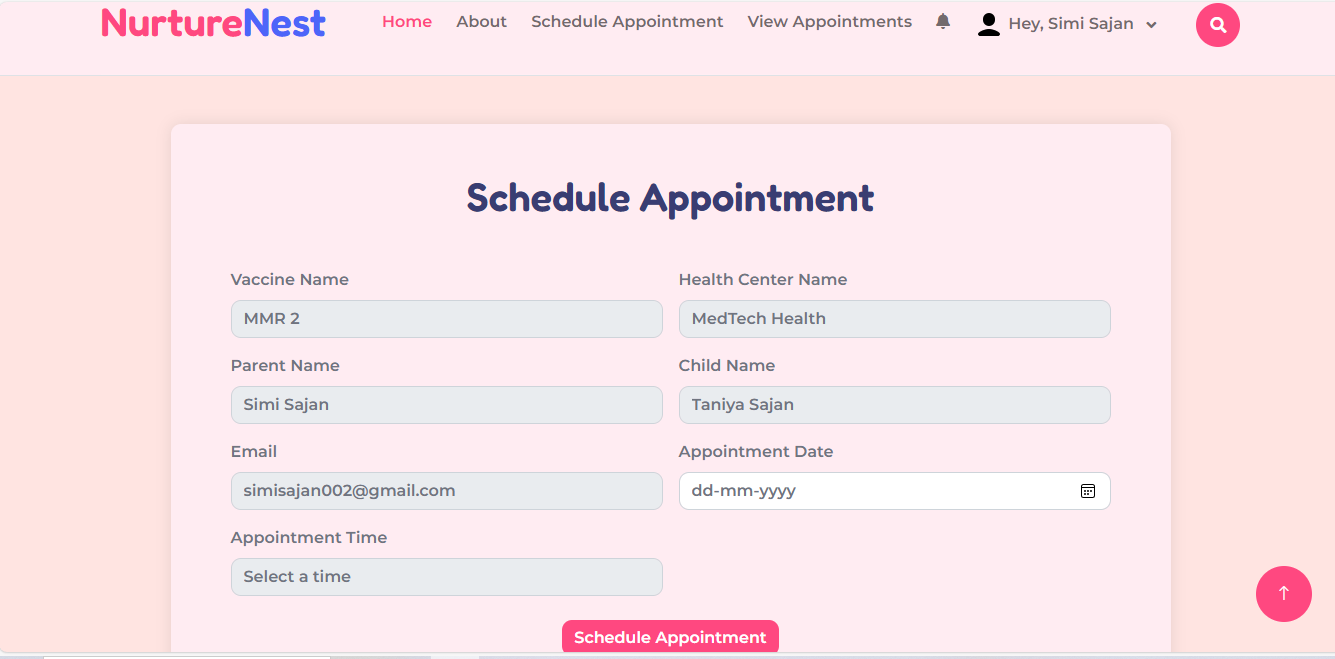
****

Fig:1 : Vaccine Scheduling Page

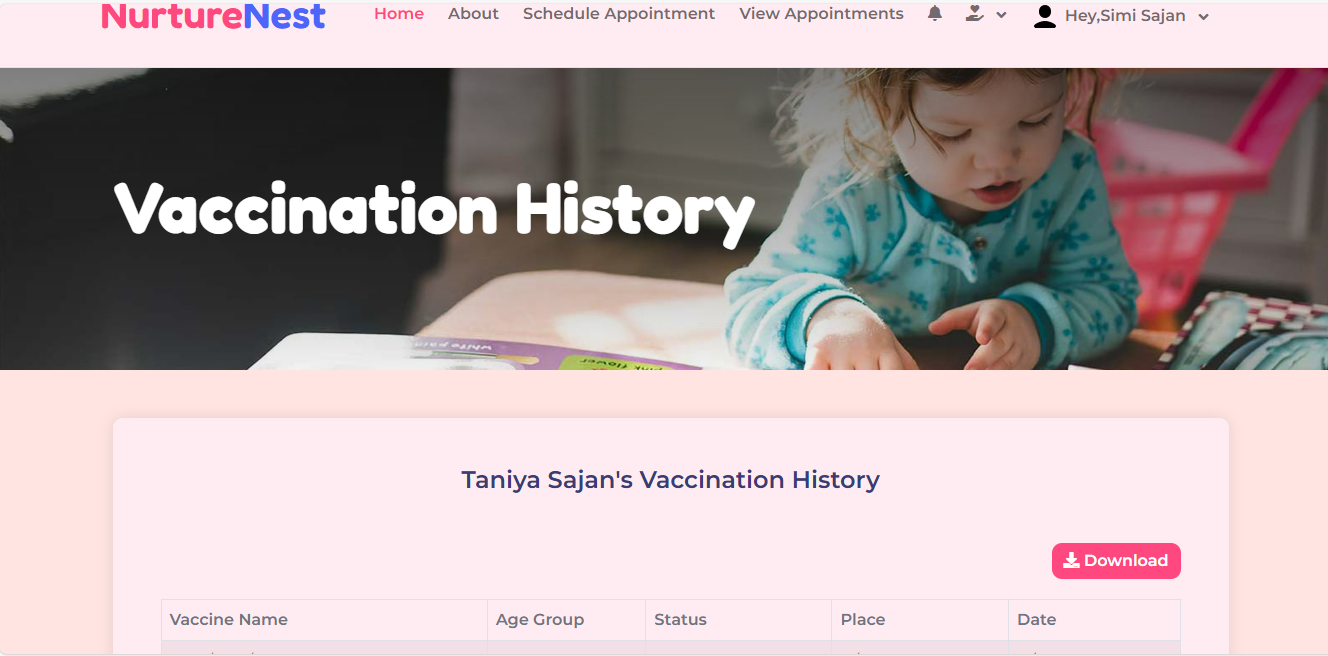
****

Fig:1 : Vaccination History Page

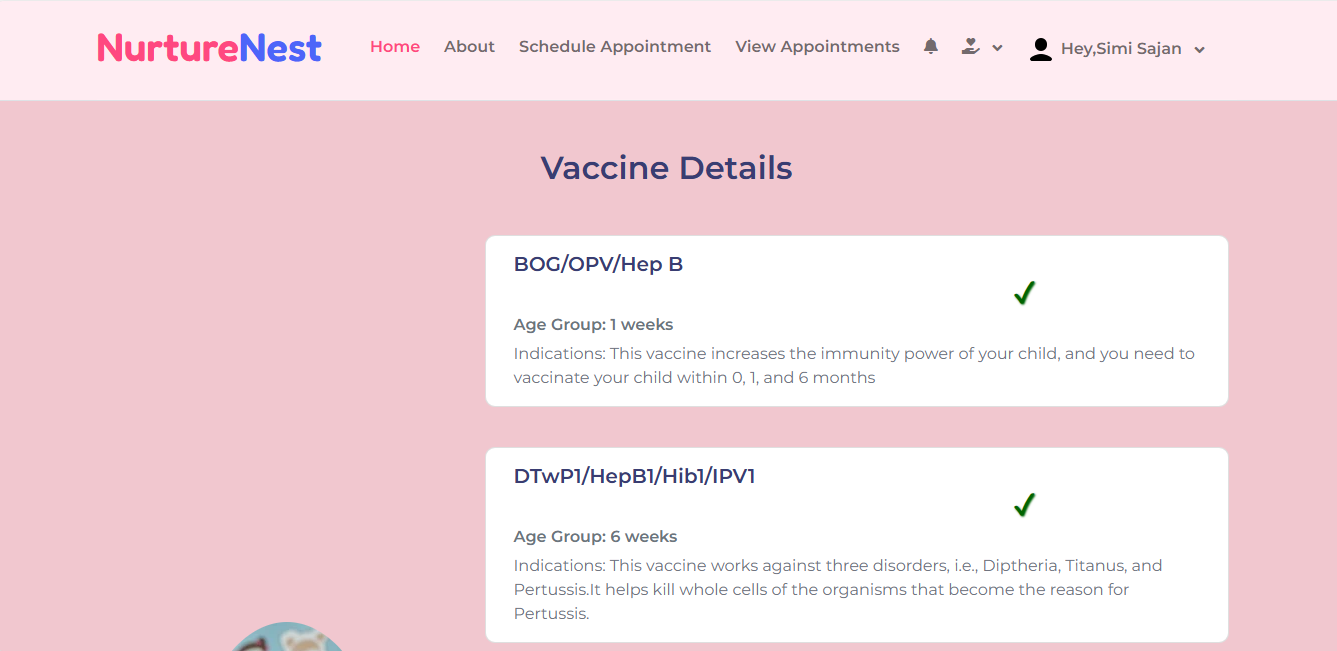
****

Fig:1 : Vaccine Chart

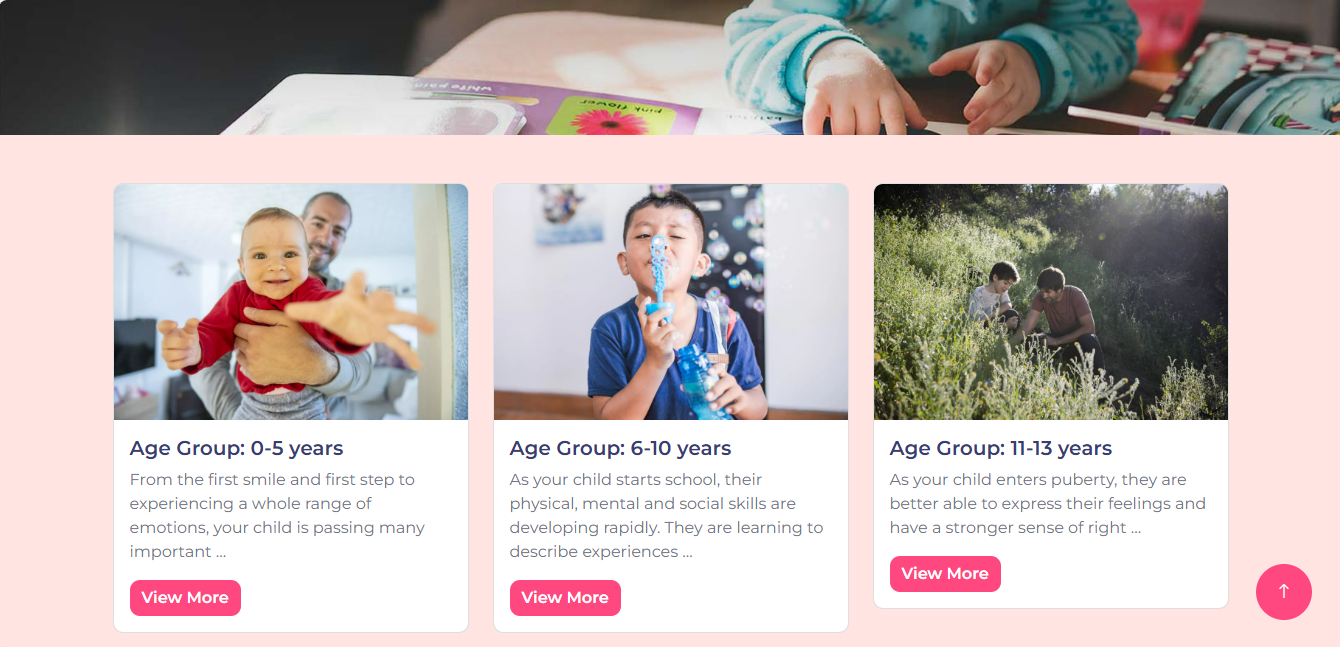
****

Fig:1 : Mental Health Page

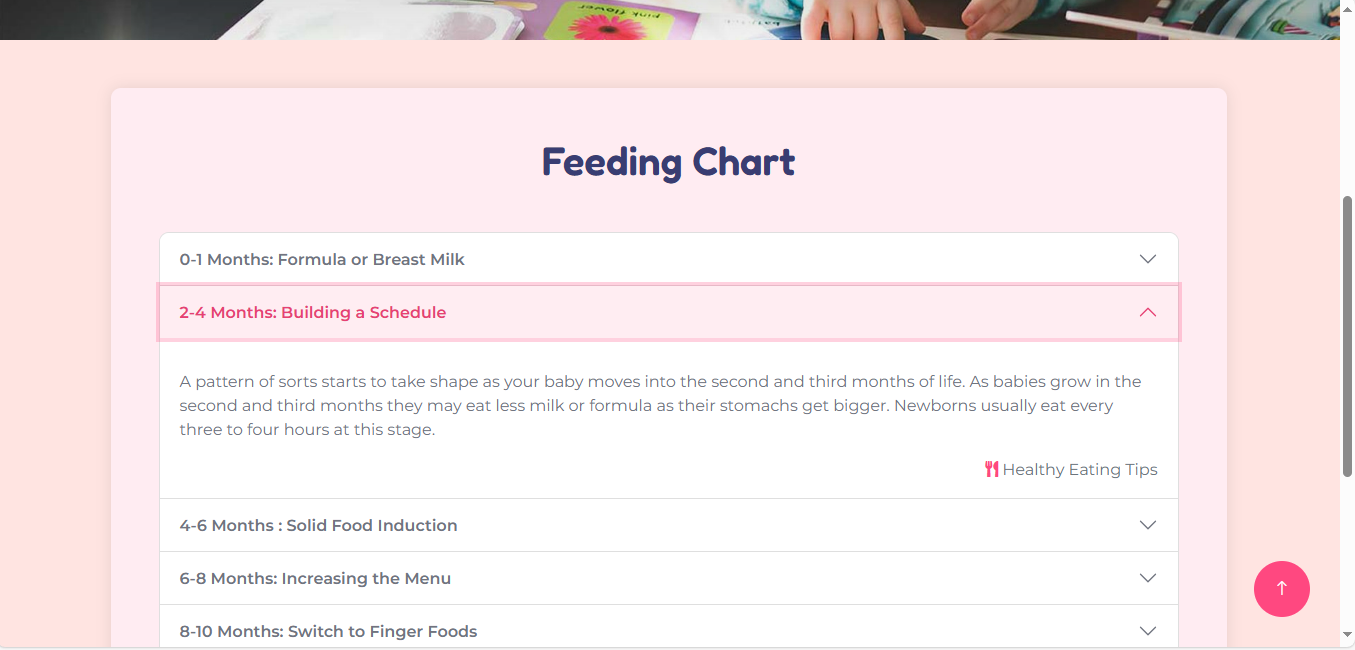
****

Fig:1 : Feeding Chart Page