*Anjuman-I-Islam’s*

#### M. H. SABOO SIDDIK COLLEGE OF ENGINEERING

8, Saboo Siddik Polytechnic Road, Byculla, Mumbai, Maharashtra 400008

**DEPARTMENT OF INFORMATION TECHNOLOGY**

**ITM601 Mini Project – 2 B for**

**ML Based Project**

# REPORT

**Title of the Project**

**Diabetics Prediction System**

**Supervisor/Guide**

### Dr. Irfan Landge

**REV - 2019 ‘C’ Scheme**



**University of Mumbai Academic Year (2024 -25)**

## CERTIFICATE

This is to certify that the project entitled “**Diabetics Prediction System**” is a bona fide work of “**Naved Sutar” (221455),** “**Saad Ansari” (221458)**, “**Hujaifa Shaikh” (221444) ,** “**Adeen Momin” (221426)** submitted to the University of Mumbai in partial fulfilment of the requirement for the ITM601 Mini Project – 2 B for ML based projectof the 6thSemester in **Department of Information Technology**.

(Dr.Irfan Landge) Supervisor/Guide

(Er. Farzana Khan) Mini Project Co-ordinator

(Dr. Zainab Mirza) Head of Department

## MINI PROJECT REPORT APPROVAL

This project report entitled “**Diabetics Prediction System**” by “**Naved Sutar**”, “**Saad Ansari**”, “**Hujaifa Shaikh**”, “**Adeen Momin**”, is approved for the ITM601 Mini Project – 2 B for ML based project of the 6th Semester.

Examiner

…………………………………………

Date: Place:

Mumbai

## DECLARATION

We declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Naved Sutar,221455

…………………………………………

Adeen Momin,221426

…………………………………………

Saad Ansari,221458

…………………………………………

Hujaifa Shaikh,221444

…………………………………………

Date:

# Abstract

Diabetes is a prevalent chronic disease that requires early diagnosis to prevent severe health complications. This project presents a Diabetes Prediction System based on Machine Learning, developed using Python and its relevant libraries. The system utilizes medical attributes such as Pregnancies, Glucose, Blood Pressure, Skin Thickness, Insulin, BMI, Diabetes Pedigree Function, and Age to predict diabetes risk. The model is trained on the PIMA Indian Diabetes Dataset, following a structured approach that includes data preprocessing, feature selection, and model training. Various classification algorithms like Logistic Regression, Decision Tree, Random Forest, and SVM were tested to identify the most accurate model. Techniques such as data normalization and hyperparameter tuning were applied to improve performance. Users input their medical data, and the trained model provides a binary classification result indicating diabetes risk. The model’s performance is evaluated using accuracy, precision, recall, and F1-score. This system aims to assist in early diabetes detection, enabling timely medical intervention and lifestyle adjustments.

**TABLE OF CONTENT**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Content** | **Page No.** |
| **1.** | **Abstract** |  |
| **2.** | **Introduction** | **1** |
| **3.** | **Methodology** | **4** |
| **4.** | **Design / Implementation** | **6** |
| **5.** | **Hardware and Software Requirements (SRS)** | **10** |
| **6.** | **Testing / Result and Analysis** | **12** |
| **7.** | **Time Line Chart** | **18** |
| **8.** | **Conclusion** | **19** |
| **9.** | **Future Scope** | **20** |
| **10.** | **References** | **22** |

**Chapter 1**

**Introduction**

Diabetes is one of the most prevalent and serious chronic diseases worldwide, affecting millions of people across different age groups. It occurs when the body either does not produce enough insulin or cannot effectively use the insulin it produces, leading to high blood sugar levels. If left undiagnosed or unmanaged, diabetes can cause severe health complications, including heart disease, kidney failure, nerve damage, and vision problems. Early detection and timely intervention are crucial in preventing these complications. The increasing number of diabetes cases globally has created an urgent need for efficient diagnostic tools that can help individuals monitor their health status and take necessary precautions before the condition worsens.

In recent years, Machine Learning (ML) has emerged as a powerful tool in the healthcare industry, enabling predictive analytics and early disease diagnosis. Traditional methods of diagnosing diabetes rely on laboratory tests and medical evaluations, which, although accurate, may not always be accessible or timely. Machine learning models, on the other hand, leverage vast amounts of medical data to detect patterns and predict the likelihood of diabetes with high accuracy. This project, Diabetes Prediction System, is a machine learning-based solution developed entirely in Python using various ML libraries. The system predicts whether an individual is likely to have diabetes based on certain medical parameters. These parameters include Pregnancies, Glucose, Blood Pressure, Skin Thickness, Insulin, BMI, Diabetes Pedigree Function, and Age. By analysing this input data, the trained model generates an output indicating the probability of diabetes.

The Diabetes Prediction System follows a structured approach involving data preprocessing, feature selection, model training, and evaluation. The dataset used for this project is the PIMA Indian Diabetes Dataset, a well-known dataset in medical research that contains relevant patient health records. Various machine learning algorithms, such as Logistic Regression, Decision Tree, Random Forest, and Support Vector Machine (SVM), have been explored to determine the most accurate model. The model is optimized using techniques like data normalization and hyperparameter tuning to improve performance and ensure reliable predictions. The implementation of different algorithms allows for a comparative analysis, ensuring that the most effective approach is selected for diabetes prediction.

One of the primary advantages of this system is its ability to provide a simple and user-friendly interface where users can input their medical details and receive a real-time prediction regarding their diabetes risk. Unlike traditional diagnostic methods, this system eliminates the need for expensive medical tests and doctor visits, making it more accessible to a broader population. Additionally, the performance of the model is evaluated using key metrics such as accuracy, precision, recall, and F1-score, ensuring its effectiveness in real-world applications. The ability to predict diabetes at an early stage can help individuals make informed lifestyle choices and seek medical advice promptly, ultimately reducing the risk of severe health complications.

With the increasing prevalence of diabetes worldwide, this machine learning-based approach offers a valuable tool for early detection. By integrating predictive analytics into healthcare, this system can assist individuals and medical professionals in making informed decisions regarding diabetes risk and management. The use of AI-driven healthcare solutions has the potential to revolutionize the medical field by providing cost-effective, efficient, and reliable disease prediction systems. This project not only demonstrates the power of machine learning in medical diagnostics but also highlights its role in enhancing preventive healthcare measures, contributing to better health outcomes for people at risk of diabetes.

**Chapter 2**

**Methodology**

The Diabetes Prediction System is developed using Machine Learning (ML) techniques to predict whether an individual has diabetes based on key medical parameters. The methodology of this project follows a structured approach, consisting of multiple stages, including data collection, preprocessing, model selection, training, evaluation, and deployment. Each phase plays a crucial role in ensuring the accuracy and efficiency of the system.

**1. Data Collection**

The dataset used for this project is the PIMA Indian Diabetes Dataset, which is widely used in medical research for diabetes prediction. It contains 768 records with 8 medical parameters:

* Pregnancies – Number of times the patient has been pregnant.
* Glucose Level – Blood glucose concentration.
* Blood Pressure – Diastolic blood pressure level.
* Skin Thickness – Thickness of skin fold in mm.
* Insulin Level – Insulin concentration in blood.
* BMI (Body Mass Index) – A measure of body fat based on height and weight.
* Diabetes Pedigree Function – Likelihood of diabetes based on family history.
* Age – Age of the individual.

Each data entry also includes a target variable (Outcome), which indicates whether the individual has diabetes (1) or not (0).

**2. Data Preprocessing**

Before training the machine learning model, data preprocessing is performed to clean and prepare the dataset. The key preprocessing steps include:

* Handling Missing Values: Any missing values in the dataset are either removed or replaced using techniques like mean, median, or mode imputation.
* Data Normalization & Scaling: Since different features have different scales, normalization (Min-Max Scaling) or standardization (Z-score normalization) is applied to bring all values within a similar range.
* Feature Selection: Correlation analysis is conducted to determine the most significant features contributing to diabetes prediction.
* Data Splitting: The dataset is split into training (80%) and testing (20%) subsets to evaluate model performance.

**3. Model Selection and Training**

Various Supervised Machine Learning algorithms are explored for model training. The models used in this project include:

* Logistic Regression – A simple linear model used for binary classification.
* Decision Tree Classifier – A tree-based model that makes decisions based on feature splits.
* Random Forest Classifier – An ensemble learning method that improves accuracy by combining multiple decision trees.
* Support Vector Machine (SVM) – A classification algorithm that finds the best hyperplane for separating data points.
* K-Nearest Neighbors (KNN) – A distance-based algorithm that classifies new data based on the closest training examples.

Each model is trained using the training dataset, and hyperparameter tuning is applied to improve performance. Techniques such as Grid Search CV and Random Search are used to find the optimal model parameters.

**4. Model Evaluation**

After training the models, their performance is evaluated using key metrics:

* Accuracy – Measures the overall correctness of the model.
* Precision – Measures how many of the predicted diabetic cases are actually diabetic.
* Recall (Sensitivity) – Measures the ability of the model to detect actual diabetic cases.
* F1-Score – A harmonic mean of precision and recall, balancing false positives and false negatives.
* Confusion Matrix – Provides insight into the number of true positives, false positives, true negatives, and false negatives.

The best-performing model is selected based on these evaluation metrics, ensuring that it provides reliable predictions for diabetes detection.

**5. Deployment Using Streamlit**

Once the optimal model is selected, it is deployesd as a web-based application using Streamlit, a Python framework for creating interactive UI applications. The system provides a user-friendly interface, allowing users to input their medical details, and the trained model processes this data to give a binary classification result ("Diabetic" or "Non-Diabetic").

Key Features of the Streamlit Application:

* User Input Form – Users enter their medical details through an interactive web form.
* Model Processing – The trained machine learning model processes the input data and predicts the diabetes status.
* Result Display – The prediction result is displayed in a simple and understandable format.
* Data Visualization – Graphical representations (such as bar charts and histograms) help users understand diabetes trends and model performance.

**Chapter 3**

**Design / Implementation**

Design and Implementation of Online Nursery Management System

* **System Architecture:** The Online Nursery Management System will be designed as a web-based application utilizing a client-server architecture.
  + Client Side:
    1. A web interface for nursery staff, management, and customers to access the system.
    2. An intuitive dashboard providing easy navigation and data visualization related to plant inventory, sales, and customer interactions.
    3. Authentication and authorization mechanisms to ensure secure access to the system.

#### Server Side:

* + A backend server built using Express.js to handle data processing, storage, and retrieval.
  + An application layer implementing business logic to manage inventory, sales, and customer relationships.
  + A database management system to efficiently store plant records, sales data, and user information.
* **Database Design:** The database will be structured to effectively manage plant inventory, customer data, and sales records.
  + Tables:
    1. Plant Table: Stores information about plant species, including ID, name, care requirements, and health status.
    2. Inventory Table: Contains details about stock levels, growth stages, and location within the nursery.
    3. Sales Table: Records sales transactions, including customer details and purchased items.
    4. User Table: Stores user credentials and access permissions for nursery staff and manage.
* **User Interface Design:** The user interface will be designed to be intuitive and user-friendly, catering to the needs of nursery staff, management, and customers.
  + Admin Interface:
    1. A dashboard displaying inventory statistics, sales performance, and important notifications.
    2. Forms for data entry and management of plant records, including updates on care schedules.
    3. Access to sales reports, customer interactions, and order processing.
    4. Tools for analyzing plant growth trends and inventory levels.
    5. Communication features to facilitate interaction with customers regarding orders and inquiries.

#### Implementation Technologies:

* + Frontend: HTML, CSS, JavaScript.
  + Backend: Node.js with Express.js
  + Database: MongoDB
  + Authentication: JSON Web Tokens (JWT) for secure user authentication and authorization.

#### Testing and Quality Assurance:

* + Comprehensive testing will be conducted at each stage of development, including unit testing, integration testing, and user acceptance testing.
  + Quality assurance measures will ensure that the system meets functional requirements, performance benchmarks, and security standards.

#### Deployment:

* + The Online Nursery Management System will be deployed on a secure web server with robust measures for data protection and access control.
  + Continuous monitoring and maintenance will be performed to ensure system stability, reliability, and scalability.

**Flowchart: -**

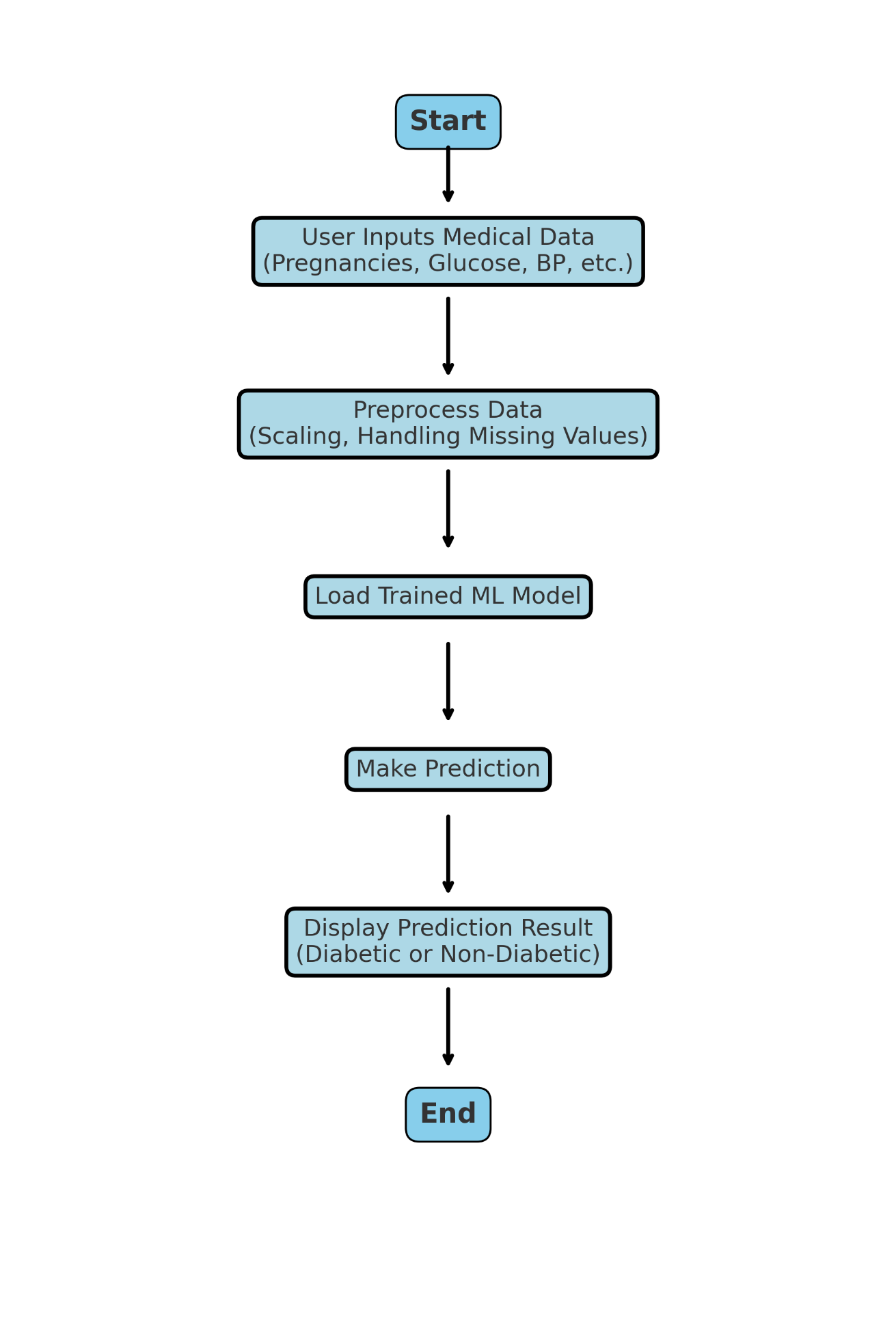
****

Fig 2: Flowchart

**Chapter 4**

**Hardware and Software Requirements (SRS)**

**Hardware Requirements:**

The hardware specifications needed for running this project efficiently are as follows:

* Processor: Intel Core i3 or higher (Recommended: i5/i7 or AMD equivalent)
* RAM: Minimum 4GB (Recommended: 8GB or more for faster processing)
* Storage: At least 20GB of free space (Recommended: SSD for faster execution)
* Graphics Card: Not required, but a basic integrated GPU is sufficient
* Display: Minimum 1366x768 resolution
* Network: Internet connection for installing required Python libraries

**Software Requirements:**

* Operating System:
  + Windows 10/11, macOS, or Linux (Ubuntu recommended)
* Programming Language:
  + Python 3.7 or higher
* Python Libraries (Required for Machine Learning and Web Deployment):
  + NumPy – For numerical computations
  + Pandas – For data manipulation and preprocessing
  + Scikit-learn – For implementing Machine Learning models
  + Matplotlib & Seaborn – For data visualization
  + Streamlit – For building and deploying the web application
* Development Environment:
  + Jupyter Notebook / VS Code / PyCharm
* Dataset:
  + **PIMA Indian Diabetes Dataset** (Downloaded from Kaggle or UCI Machine Learning Repository)

**Chapter 5**

**Testing / Result and Analysis**

**Chapter 6**

**Time Line Chart**

The main aim of timeline chart is to make it easier to capture a process as a sequence of events and – as such – to understand all dependencies within a project, estimating the time that it takes every step or money that will be needed to complete it. Timeline charts are commonly used to oversee team workload and grasping the overall progress in time. Additionally, timeline chart can highlight critical events and display milestones, and visualize the ownerships of particular stages or tasks.

Fig 11: Timeline chart

**Chapter 7**

**Conclusion**

The Diabetes Prediction System effectively leverages machine learning to provide a reliable and accessible solution for early detection of diabetes based on key medical parameters. By automating the prediction process through a user-friendly web interface built with Streamlit, it empowers individuals to assess their health risk conveniently from anywhere.

Healthcare professionals and users benefit from this system through quick, data-driven insights that aid in timely decision-making and preventive measures. The use of established medical datasets and ML algorithms ensures accurate predictions while maintaining simplicity in user experience.

Additionally, the system promotes awareness about health monitoring and encourages proactive health management. It supports sustainable healthcare by reducing the need for physical consultations for initial screenings and helps prioritize patients who need immediate medical attention.

The Diabetes Prediction System is a practical and modern healthcare solution that bridges technology with medical diagnostics, making predictive healthcare more scalable, accessible, and impactful in everyday life.

**Chapter 8**

**Future Scope**

The Diabetes Prediction System project provides a robust foundation for leveraging machine learning in healthcare diagnostics. While the current system accurately predicts diabetes based on key medical parameters, there are several avenues for future development to enhance its scope, usability, and real-world impact. Below are some potential areas for expansion and improvement:

#### Functionality and Features:

* + Integrate additional medical parameters such as cholesterol levels, physical activity, dietary habits, and genetic factors to improve prediction accuracy.
  + Implement real-time data integration from wearable health devices or health apps for continuous monitoring.
  + Add a multi-model comparison interface allowing users or doctors to see predictions from multiple algorithms.

#### Enhanced Communication and Collaboration:

* + Include a chatbot or AI assistant for guiding users through the input process or interpreting results.
  + Enable integration with hospital databases or health information systems for direct data exchange and better patient management.
  + Provide a log feature to store user history and track diabetes risk progression over time.

.

#### Mobile Application Development:

* + Develop a cross-platform mobile app version of the system to allow users to input and track their health data anytime, anywhere.
  + Integrate features like daily reminders for health check-ins or alerts when high-risk thresholds are crossed.

#### Customization and Personalization:

* + Allow users to create personalized health dashboards that display metrics such as BMI trends, glucose history, and lifestyle recommendations.
  + Customize prediction suggestions based on age groups, gender, or geographic health trends for more targeted insights.

**Chapter 9**

**References**

* **MongoDB Documentation**  
  MongoDB Inc. (2023). MongoDB Documentation. Retrieved from <https://docs.mongodb.com>
* This reference provides detailed information on how MongoDB, a NoSQL database, is used in the MERN stack for managing and storing product data in the online nursery management system.
* **React Official Documentation**  
  Meta Platforms, Inc. (2023). React – A JavaScript library for building user interfaces. Retrieved from https://reactjs.org/docs/getting-started.html
* This source explains the use of React for developing the dynamic front-end of the system, allowing users to explore products and manage their cart in real-time.
* **Node.js Documentation**  
  OpenJS Foundation (2023). Node.js v20.x Documentation. Retrieved from <https://nodejs.org/en/docs>
* Node.js is utilized to handle the server-side logic of the project. This reference explains its usage in processing requests, handling authentication, and managing product orders.
* **Express.js API Reference**  
  Express.js Foundation (2023). Express 5.x Documentation. Retrieved from https://expressjs.com/en/5x/api.html
* Express.js is used in the system's back-end to create APIs that handle user login, product listing, and cart management. This document provides an in-depth look at how Express helps build RESTful APIs.
* **JavaScript: The Definitive Guide**  
  Flanagan, D. (2020). JavaScript: The Definitive Guide (7th Edition). O'Reilly Media.
* This book provides essential insights into JavaScript, which is used across the project for front-end and back-end scripting, including the user interface interactivity and server-side logic.
* **E-commerce UX Design**  
  Baymard Institute (2023). E-Commerce UX Guidelines. Retrieved from https://baymard.com/research/ecommerce-ux
* This research report covers best practices for designing user-friendly e-commerce platforms, contributing to the design of the explore page, search functionality, and user interactions within the nursery management system.