

A Practical Training Paper on Machine Learning-Driven Web Development Using Multiple Disease Prediction System

¹Savita Sahu,²Shivam Kumar Gupta, ³Sahil Meshram, ⁴Yatharth Sonteke, ⁵Shivam Sahu

¹Assistant Professor, ^{2,3,4,5}Bachelor Of Technology

^{1,2,3,4,5}Department Of Computer Science &Engineering Government Engineering College
Bilaspur, C.G.

ABSTRACT

This project involves developing a comprehensive Multiple Disease Prediction System using modern web development technologies such as Next.js, Drizzle ORM, Neon for database management, Recharts for visualizations, and Clerk for authentication. The application is designed to help users efficiently manage and track their expenses with a dynamic and interactive user interface. The backend leverages Next.js server functions and integrates seamlessly with the Neon database via Drizzle ORM, ensuring secure, scalable, and efficient data management. The front-end, built with ReactJS, delivers a seamless and engaging user experience, allowing users to create, update, and delete expense entries, categorize expenses, and analyze their spending habits using intuitive visualizations. Key features of the application include recording expenses with details such as date, category, and amount; displaying expenses in an organized list format; and analyzing spending patterns through interactive bar chart powered by Recharts. Users can filter expenses by date range, category, or amount to gain deeper insights into their financial behavior. Authentication is powered by Clerk, ensuring secure and personalized access to user data. The project also incorporates responsive design principles, ensuring accessibility across devices, including desktops, tablets, and smartphones. With its robust architecture, secure database integration, and insightful visualizations, this Multiple Disease Prediction System serves as an ideal solution for individuals seeking to enhance financial awareness and take control of their spending efficiently.

KEYWORDS: Machine Learning, Web Development, Python, JavaScript, MATLAB, Responsive Front-End, DOM Manipulation, Asynchronous Programming, Tourism Website.

INTRODUCTION

In today's fast-paced world, managing personal finances has become more critical than ever. Many individuals struggle with tracking their expenses, organizing financial data, and gaining actionable insights to improve their financial habits. Recognizing these challenges, the Multiple Disease Prediction System was designed to provide users with a comprehensive, intuitive, and intelligent platform for managing their finances. This application combines cutting-edge technologies with a user-centric approach to simplify expense tracking, provide insightful analytics, and foster better financial discipline.

At the heart of this project is a focus on data-driven insights and advanced analytics. Leveraging Drizzle ORM for seamless database integration and Recharts for dynamic data visualization, the application allows users to monitor their spending habits in real time. By categorizing expenses and offering interactive visual summaries, users can identify patterns, optimize budgets, and make informed financial decisions effortlessly.

Another standout feature is the application's ease of deployment and scalability. By using Next.js server functions and integrating with the Neon database, the backend is not only robust and efficient but also scalable to meet growing user demands. Developers benefit from detailed documentation, tutorials, and technical support, making the platform simple to integrate and maintain.

In essence, this application represents a transformative approach to personal finance management. By combining intuitive design, powerful analytics, and advanced technologies

TOOLS & TECHNOLOGY

PROJECT DESIGN

The Expense Tracking Application relies on two key data sources: the database and the web scrapers. The database stores user interaction data, including click events and other user activities, allowing for efficient management and CRUD operations via an admin panel. The web scrapers are designed to track almost every event the user performs, sending this data to the database for analysis. This data helps provide insights into user behaviour and website performance.

The frontend API serves as the interface between the front end and the backend, ensuring smooth communication. The front-end of the application is a dynamic single-page interface, developed using React, Next.js, and Tailwind CSS. It includes various components like interaction logs, analytics, user signup, login, and admin pages, offering a seamless and engaging user experience.

TECHNOLOGY USED

Visual Studio Code (VS Code)

Visual Studio Code (VS Code) is the main code editor for this project. It provides powerful features like syntax highlighting, IntelliSense for code completion, debugging support, and Git integration, making it ideal for both frontend and backend development. Its extensive library of extensions enhances productivity and supports various programming languages, including JavaScript, HTML, and CSS

Next.js

Next.js is a React framework that enables both static and dynamic website rendering. It provides features such as server-side rendering (SSR) and static site generation (SSG), making it a powerful choice for building highly performant web applications. Next.js is known for its seamless integration with React and its ability to optimize applications for better performance and SEO.

Tailwind CSS

Tailwind CSS is a utility-first CSS framework that provides pre-built classes for creating custom designs. It allows developers to style elements directly within the HTML by applying utility classes, offering high flexibility and reducing the amount of custom CSS code needed. Tailwind is known for its efficiency and ability to streamline the development process.

Node.js

Node.js is a runtime environment that allows JavaScript to be executed on the server side. It uses a non-blocking, event-driven architecture, making it highly efficient for building scalable network applications. With Node.js, backend services for the application are fast, and it supports a wide array of libraries and tools.

Express.js

Express.js is a minimal web application framework for Node.js that simplifies the process of building web servers and APIs. It provides a robust set of features for routing, handling HTTP requests, and middleware integration. Express.js enables the rapid development of backend services, making it ideal for RESTful APIs in this application.

Drizzle ORM

Drizzle ORM is an Object-Relational Mapping (ORM) tool that simplifies the interaction between the application and the database. It allows for easy management of database queries and

helps maintain clean, readable code. Drizzle ORM is designed to work efficiently with modern JavaScript frameworks like React and Next.js.

LITERATURE REVIEW

The need for efficient and user-friendly Multiple Disease Prediction solutions has grown significantly in recent years, driven by the increasing complexity of personal and business financial management. Early financial management systems were primarily manual, relying on physical ledgers and spreadsheets, which were prone to errors, time-consuming, and lacked the analytical capabilities necessary for modern users.

Recent advancements have shifted focus to digital tools, integrating automation, real-time updates, and user-centric designs. Numerous studies have explored the effectiveness of mobile and web-based applications for tracking expenses. These applications leverage features like intuitive interfaces for data entry, dynamic categorization, and advanced analytics to help users identify spending patterns, optimize budgets, and achieve financial goals.

Researchers have also highlighted the importance of scalable and efficient backend systems. Database management frameworks such as Drizzle ORM, paired with platforms like Neon, enable secure and efficient handling of user data. These technologies ensure that applications can accommodate growing user bases without compromising performance or security.

Additionally, predictive analytics and machine learning are emerging as pivotal tools for enhancing the utility of expense trackers by providing forecasts and personalized financial recommendations.

In conclusion, the evolution of expense tracking applications has been shaped by advancements in technology and an increasing demand for smarter financial tools. Current research underscores the need for secure, scalable, and user-friendly solutions that empower users to manage their finances effectively, paving the way for continuous innovation in this domain.

METHODOLOGY

AGILE METHODOLOGY

The Agile methodology is a way to manage a project by breaking it up into several phases. It involves constant collaboration with stakeholders and continuous improvement at every stage. Once the work begins, teams' cycle through a process of planning, executing, and evaluating. Continuous collaboration is vital, both with team members and project stakeholders.



Fig 3.1. Agile Methodology

AGILE MODEL LIFECYCLE

The Agile model lifecycle is an iterative and incremental process that emphasizes flexibility, collaboration, and customer satisfaction. It breaks down the development process into smaller cycles, each delivering a functional product increment. The key phases include:

Planning: The planning phase involves defining the project objectives, gathering requirements, and prioritizing tasks into manageable user stories or sprints. Stakeholders collaborate to ensure a clear understanding of the project goals and align expectations for the deliverables.

Design: In the design phase, the focus is on creating wireframes, technical blueprints, and prototypes. This ensures that the architecture and user interface are well-structured and support the intended functionalities, laying a strong foundation for development.

Develop: During the development phase, the team builds the product incrementally by implementing the planned features. Continuous integration and frequent collaboration ensure that the product evolves efficiently and aligns with user expectations.

Testing: The testing phase focuses on identifying and fixing issues in the product. Teams conduct various tests, such as unit, integration, and acceptance tests, to ensure functionality, reliability, and compatibility with the defined requirements.

Deploy: In the deployment phase, the product is moved to a staging or live environment. This step ensures the software is accessible to users or stakeholders for further evaluation, with necessary configurations and updates.

Review: The review phase involves collecting feedback from stakeholders and users to identify areas for improvement. The backlog is updated based on the insights gained, preparing for subsequent iterations.

Launch: The launch phase marks the product's official release into the production environment. It includes monitoring the product's performance, ensuring stability, and providing necessary documentation and support to end users.

BLOCK DIAGRAM

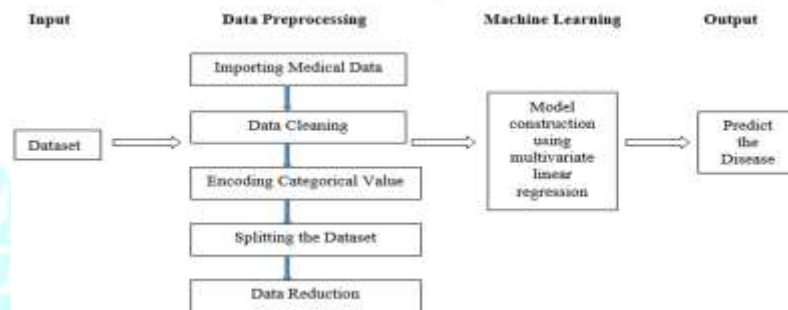


Fig 3.2 Block Diagram

DATA FLOW DIAGRAM

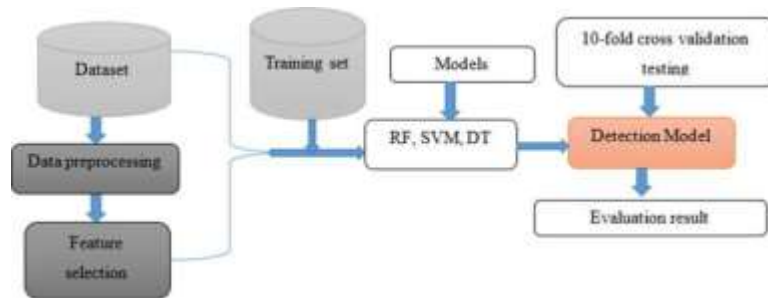


Fig 3.3 DFD Level 0

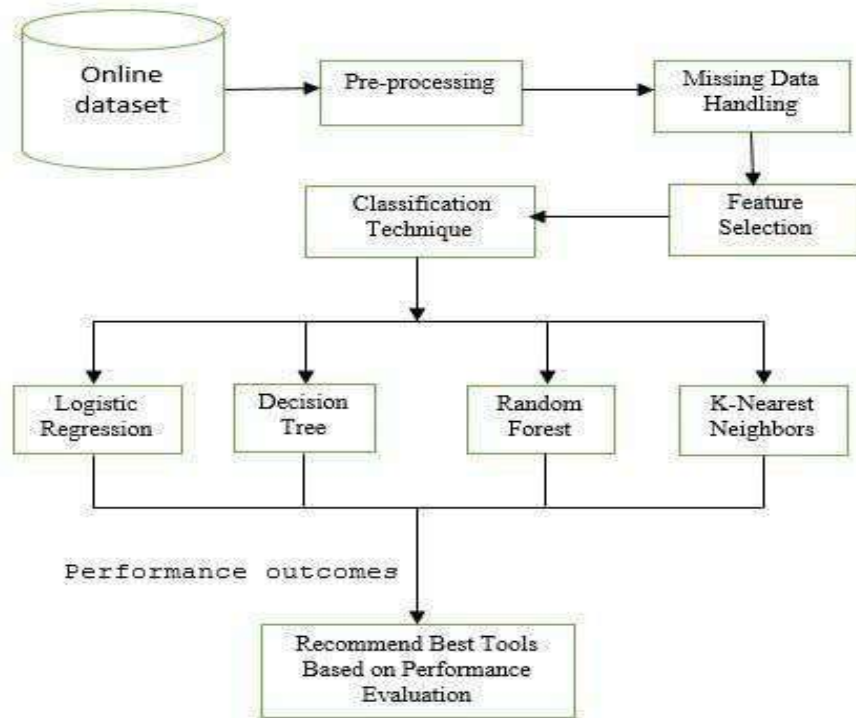


Fig 3.4 DFD Level 1

IMPLEMENTATION AND RESULT

IMPLEMENTATION JOURNEY

The implementation of the Multiple Disease Prediction System involves several key steps to ensure smooth user experience, secure data management, and seamless integration of features:

User Authentication Setup:

Clerk Integration: The first step in the implementation is integrating Clerk for secure user authentication. This allows users to log in through multiple methods like email, Google, or Facebook.

Session Management: Clerk also handles the session management, ensuring personalized access for each user, such as storing user preferences and ensuring security.

Data Security: Encryption standards are implemented to protect sensitive data, ensuring that unauthorized access is prevented and maintaining compliance with global privacy standards (e.g., GDPR).

Data Entry and User Interface:

Intuitive Form Design: A simple, user-friendly data entry form is created to allow users to input details of their expenses quickly. Fields include amount, date, category, and optional notes.

Data Validation: Real-time checks are implemented to ensure that the data entered is accurate, reducing errors like missing values or invalid amounts.

Ease of Use: The form's design is streamlined for ease of use, allowing users to log their expenses quickly and effectively.

Categorization and Budget Management:

Automatic Categorization: The system automatically categorizes expenses into predefined categories (e.g., groceries, rent) to make it easy for users to track spending. Users can also create custom categories.

Budget Monitoring: Users can set spending limits for each category. Notifications are triggered when they approach or exceed these limits to help them maintain financial discipline.

Customizable Settings: Users can adjust their budgets and categories based on their personal financial goals.

RESULTS

User Authentication:

The authentication system works flawlessly, offering multiple secure login methods. Data privacy measures have been successfully implemented, ensuring that users' sensitive financial information is protected and compliant with global standards like GDPR.

Data Entry Efficiency:

The data entry system is user-friendly and efficient. The real-time validation ensures that users can log expenses accurately without facing errors, enhancing the overall user experience.

Budget Control and Notifications:

The categorization and budget monitoring tools are highly effective. Users have successfully set and monitored spending limits, with real-time notifications helping them stay on track with their financial goals. This feature has enhanced user engagement by encouraging more proactive budget management.

Visual Insights and Financial Management:

The interactive charts and filtering options have received positive feedback from users. The ability to visualize expenses through bar charts, pie charts, and line graphs has helped users identify spending patterns and make data-driven decisions to optimize their budgets.

Data Security:

Data security has been rigorously implemented, and users have expressed confidence in the protection of their personal information. The system's adherence to privacy regulations ensures user trust, while regular backups safeguard against data loss.

IMPLEMENTED PROJECT SCREENSHOTS:

Fig 4.1. Landing Page

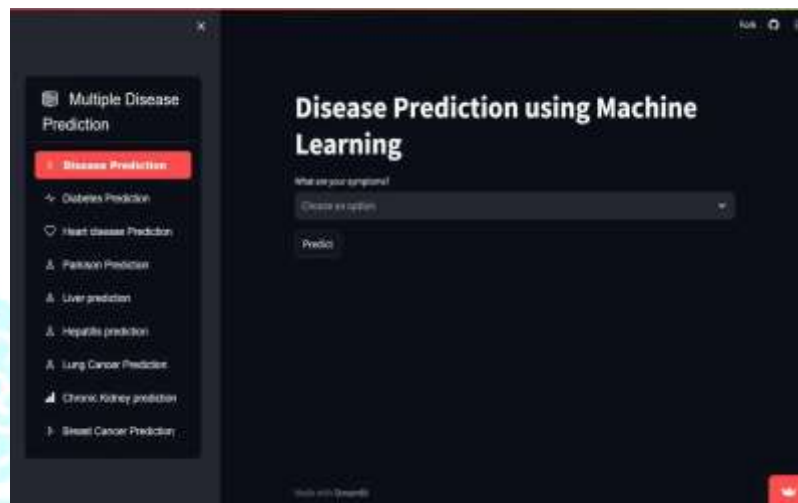


Fig 4.2. Input Page



FUTURE SCOPE

The Multiple Disease Prediction System has vast potential for future development, expanding its capabilities to meet the growing needs of users in various financial management areas. The following outlines potential advancements and features that can further enhance the app's utility:

Personal Health Manage

AI-based Budgeting: Future updates could integrate AI to automatically categorize expenses, analyze spending patterns, and offer personalized budgeting advice.

Automated Savings Goals: By linking the app to users' bank accounts, it could automatically allocate a portion of income toward savings goals based on income and spending behavior.

Debt Management Plans: The app can evolve to include AI-powered recommendations for structuring optimal debt repayment plans, considering interest rates, payment history, and user preferences.

Predictive Spending Insights: AI-driven insights can predict future spending trends based on past behaviors, helping users plan for upcoming expenses.

Disease Management

Shared Financial Dashboard: The app could introduce a shared dashboard for family members, providing visibility into family-wide expenses and allowing more collaborative budget management.

Family Goals and Budgeting: Future versions could allow families to set collective savings goals (e.g., for vacations or home renovations) and track progress in real time.

Child Financial Education: For families with young members, a feature could be added to help teach children about financial management through a child-friendly interface.

Accurate Disease Prediction

Advanced Tax Reporting: By integrating with accounting software or tax APIs, the app could automatically generate tax reports, categorizing deductible and non-deductible expenses.

Expense Forecasting: The app could provide future cost predictions based on historical data, helping businesses plan for the future.

CONCLUSION

The Expense Tracker web application is a comprehensive solution to the modern challenges of personal finance management. By automating the process of tracking expenses, it addresses the complexities of organizing financial data, categorizing expenditures, and analyzing spending patterns. The application leverages a modern and robust tech stack, including React, Next.js, Drizzle ORM, Neon, and Clerk, to deliver a seamless, secure, and efficient user experience.

At its core, the application's intuitive interface ensures ease of use, catering to both tech-savvy individuals and those new to financial management tools. The incorporation of real-time analytics powered by Recharts offers users actionable insights into their spending habits, enabling them to identify trends, manage budgets, and make informed financial decisions. This dynamic visualization empowers users to take control of their financial health in a meaningful and accessible way. Security and scalability are fundamental pillars of the application. Clerk's authentication services safeguard user data with robust protocols, fostering trust and reliability. Meanwhile, the integration with Neon and Drizzle ORM ensures efficient data management, supporting real-time synchronization and the capacity to scale with growing user needs. This infrastructure guarantees consistent performance even as the application accommodates more features or a larger user base. Beyond personal use, the application holds potential for broader applications, such as supporting small businesses, aiding educational initiatives, and fostering collaborative budgeting among families or teams. Its flexibility makes it a versatile tool for diverse user groups.

REFERENCES

- [1] USING DIFFERENT MACHINE Conference on Computing Methodologies and Communication (ICCMC)
- [2] Archana Singh, Rakesh Kumar, "Heart Disease Prediction Using Machine Learning Algorithms", 2020 IEEE, International Conference on Electrical and Electronics Engineering (ICE3)
- [3] A. Sivasangari, Baddigam Jaya Krishna Reddy, Annamareddy Kiran, P. Ajitha," Diagnosis of Liver Disease using Machine Learning Models" 2020 Fourth International Conference on I-SMAC
- [4] P. Krishnaiah, et al., "Predictive Modeling for Multiple Diseases Using Machine Learning with Feature Engineering," 2015.
- [5] H. Al-Mallah, et al., "Multiple Disease Prediction Using Hybrid Deep Learning Architecture," 2016.

BIOGRAPHY

1. Sahil Meshram pursuing B. Tech 7th Semester in Computer Science Engineering from Government Engineering College Bilaspur.
2. Shivam Sahu pursuing B. Tech 7th Semester in Computer Science Engineering from Government Engineering College Bilaspur.
3. Shivam Kumar Gupta pursuing B. Tech 7th Semester in Computer Science Engineering from Government Engineering College Bilaspur.
4. Yatharth Sonteke pursuing B. Tech 7th Semester in Computer Science Engineering from Government Engineering College Bilaspur.
5. Nawaz Mehmood pursuing B. Tech 7th Semester in Computer Science Engineering from Government Engineering College Bila