A Mini Project Report on Introduction to Chatbot to Individual Prakriti Using Ayurvedic App

Mrs. Amita Mishra[1], B.Harsha Vardhan[2], G.Bharath Kumar[3], MD.Sofiya[4]

Assistant Professor[1], Department of Computer Science and Engineering, Anurag University, Telangana, India.
Student[2,3,4], Department of Computer Science and Engineering, Anurag University, Telangana, India.

Abstract. The concept of Prakriti, derived from Ayurveda, is a unique blend of physical, physiological, and psychological attributes that determine an individual's constitution. Understanding one's Prakriti is essential for personalized health recommendations, lifestyle adjustments, and preventive healthcare strategies. This app helps us to know the food consumptions you need to take according to your body conditions for better health. In this app we used node js as backend and react js as our front end. We also used html css for app user interface. Existing methods for Prakriti determination rely on questionnaires, direct consultations with Ayurvedic practitioners, and physical examinations. These methods can be time-consuming, subjective, and inaccessible to a wider audience. We propose a novel chatbot solution that leverages artificial intelligence to determine an individual's Prakriti. The chatbot, designed with natural language processing capabilities, interacts with users to gather information about their physical attributes, habits, and mental state.

Keywords. Chatbot, Prakriti, Ayurvedic, Application, Health and Nutrition.

1. INTRODUCTION

Ayurveda, the ancient Indian system of medicine, is deeply rooted in the balance of the body's natural elements, which are categorized into three doshas: Vata, Pitta, and Kapha. These doshas govern physical, emotional, and mental aspects of health, and maintaining their equilibrium is key to achieving well-being. In today's fast-paced world, understanding individual health needs based on Ayurvedic principles can be challenging. To address this, our project focuses on developing a user-friendly Ayurvedic app that helps individuals assess their dosha balance and provides personalized health recommendations based on their unique constitution and daily intake.

This app is designed to empower users by integrating the traditional knowledge of Ayurveda with modern technology, offering personalized dietary and lifestyle suggestions tailored to each user's dosha. The app aims to promote health and well-being by educating users on proper nutrition and lifestyle practices according to Ayurvedic principles.

1.1Overview

The Ayurvedic app project focuses on two main components:

Dosha Assessment: The app provides a comprehensive questionnaire to determine the user's dominant dosha. Based on the user's physical, emotional, and behavioral traits, it identifies whether they are primarily Vata, Pitta, or Kapha, or a combination of these doshas.

Health and Nutrition Recommendations: Once the user's dosha is identified, the app generates personalized dietary and lifestyle recommendations. These suggestions include:

- Foods to favor or avoid based on the user's dosha.
- Daily routines and lifestyle practices to maintain dosha balance.
- Herbal remedies and Ayurvedic supplements to promote well-being.

The app's interface is designed to be intuitive and accessible, ensuring users can easily navigate through the features and track their progress. By offering tailored guidance, the app helps users achieve a balanced lifestyle in harmony with their unique Ayurvedic constitution.

1.2Objectives

1 To assess users' Ayurvedic dosha constitution

Develop a reliable questionnaire-based system that accurately identifies the user's dominant dosha (Vata, Pitta, or Kapha) based on their physical, mental, and emotional characteristics.

2 To provide personalized dietary recommendations

Offer tailored nutrition plans that align with each user's dosha, promoting balance and well-being through appropriate food choices according to Ayurvedic principles.

3 To deliver lifestyle suggestions for dosha balance

Provide daily routines, exercises, and lifestyle tips that help users maintain or restore the balance of their doshas, enhancing overall health.

4 To educate users on Ayurvedic health practices

Enhance user awareness about the significance of Ayurveda in daily life by offering insights into herbal remedies, seasonal routines, and general health tips.

5 To create an intuitive user interface

Design the app for easy navigation, ensuring users of all ages and backgrounds can access their dosha information and health suggestions with minimal effort.

6 To track users' health progress over time

Implement features that allow users to monitor changes in their dosha balance, track health improvements, and adjust their routines as needed based on the app's recommendations.

7 To integrate modern technology with traditional Ayurveda

Leverage the power of digital platforms to make ancient Ayurvedic knowledge more accessible and actionable for users in today's technology-driven world.

1.3Problem formulation

In today's fast-paced world, maintaining good health is becoming increasingly challenging due to poor dietary habits, irregular lifestyles, and rising stress levels. Despite the widespread availability of health resources, there is often a lack of personalized guidance that takes into account the unique constitution of an individual. Ayurveda, a centuries-old system of medicine, offers a holistic approach to health by focusing on the balance of the three doshas: Vata, Pitta, and Kapha. However, the traditional knowledge of Ayurveda can be complex and difficult to apply without expert consultation.

1. Key Problems

Lack of Personalized Health Guidance: Most modern health applications and services do not consider the individual’s unique body constitution or dosha imbalance, which is central to Ayurvedic health management.

Difficulty in Understanding Ayurvedic Principles: For the average user, understanding and applying Ayurvedic concepts such as doshas, dietary guidelines, and lifestyle practices can be overwhelming and confusing without expert guidance.

Inaccessibility of Ayurvedic Expertise: Access to professional Ayurvedic practitioners can be limited, expensive, or geographically unavailable for many individuals who may benefit from personalized health advice.

Limited Use of Technology in Ayurveda: Although Ayurveda is gaining popularity, there is still a lack of integration between modern technology and traditional Ayurvedic practices to make this knowledge easily accessible and actionable for the general public.

2. Problem Statement

How can we create a digital solution that effectively integrates Ayurvedic principles of health and wellness, offers personalized guidance based on an individual's dosha constitution, and provides practical recommendations for diet and lifestyle in an easy-to-understand format?

1.4Scope of the project

The Ayurvedic app project focuses on delivering a comprehensive and user-friendly platform that integrates the principles of Ayurveda to offer personalized health and wellness guidance. The app is designed to cater to a wide range of users, from those familiar with Ayurveda to those new to the concept, by providing simplified yet effective tools for understanding and applying Ayurvedic recommendations in daily life.

1. *Key components*

Dosha Assessment System:

- The app will include a questionnaire-based dosha assessment tool, which evaluates the user's physical, emotional, and mental characteristics to determine their dominant dosha (Vata, Pitta, Kapha, or a combination).
- The dosha assessment is designed to be easy to understand, with simple questions that can be completed in a few minutes.

Personalized Health Recommendations:

- Based on the user's dosha, the app will provide customized dietary and lifestyle recommendations to promote balance and well-being.
- The dietary recommendations will include foods to favor or avoid based on Ayurvedic principles.
- Lifestyle recommendations will include daily routines, exercise, and wellness practices suited to each dosha type.

User Interface and Experience:

- The app will have an intuitive user interface, allowing users to easily navigate through the dosha assessment, health recommendations, and progress tracking features. ○ A focus on simplicity and accessibility ensures that the app is usable by individuals of all ages and technological skill levels.

Health Tracking and Monitoring:

- Users will be able to track their health progress over time, monitoring how their dosha balance changes in response to the recommendations provided.
- The app will include a progress log where users can input their health data and
- symptoms to receive updated suggestions.

Educational Content:

- The app will offer educational materials on Ayurveda, including explanations of the doshas, dietary principles, seasonal routines, and Ayurvedic herbs.
- This will empower users with the knowledge to make informed decisions about their health based on Ayurvedic practices.

Future Extensions:

- The app has the potential for further development, including integration with wearable devices for health tracking, additional features such as yoga and meditation routines, and a wider range of Ayurvedic resources.
- Future updates may also include multi-language support, expanded dietary databases, and more detailed customization options for users with specific health conditions.

1.5Feasibility

1. *Technical Feasibility*

The development of an Ayurvedic app that assesses doshas and provides personalized health recommendations is technically feasible given current technologies. The key components of the app, such as dosha assessments, dietary recommendations, and user health tracking, can be implemented using standard software development tools.

- Platform: The app can be developed for mobile platforms (Android and iOS) using frameworks like Flutter or React Native, which allow for cross-platform compatibility.
- Database: A robust backend database (e.g., Firebase or MySQL) will be required to store user data, including dosha assessments, health progress, and personalized recommendations.
- AI/ML Integration (optional): Future versions of the app could incorporate AI or machine learning algorithms to offer more advanced personalized recommendations by analyzing user behavior and health data.
- User Interface: Creating an intuitive, easy-to-navigate user interface is achievable with modern UI/UX design tools like Figma or Adobe XD.

Overall, there are no significant technical barriers that would prevent the app from being developed and deployed.

2. *Economic Feasibility*

The economic feasibility of the Ayurvedic app project is highly favorable:

- Initial Development Costs: The initial costs would include app development, design, and testing. Depending on the scope of features and the team size, this could range from moderate to high. However, using cost-effective frameworks and outsourcing some of the work could reduce expenses.
- Operational Costs: Once the app is developed, operational costs will include hosting, data storage, periodic updates, and maintenance. These costs can be managed through cloud services (e.g., AWS, Google Cloud) on a scalable basis.
- Revenue Models: The app could generate revenue through:
    - Freemium model: Basic features can be offered for free, with premium features or personalized consultations offered as in-app purchases.

- Subscriptions: Monthly or annual subscriptions for access to advanced features, personalized health plans, or expert consultations.

Return on Investment (ROI): Given the growing interest in wellness and Ayurveda, the potential user base is large, which makes the investment likely to generate a good return, especially with strategic marketing.

3. *Operational Feasibility*

The operation of the app is highly feasible. Once developed, maintaining the app primarily involves monitoring user feedback, fixing bugs, and updating features. Content updates, such as new Ayurvedic tips, recipes, and seasonal recommendations, can be managed by a small team of experts.

- User Support: A basic customer support system can be implemented using chatbots and FAQs to handle common queries. More complex issues can be addressed by a small support team.
- Scalability: The app can be scaled to handle a growing user base by leveraging cloud infrastructure for hosting and database management.

4. *Market Feasibility*

The market feasibility is highly promising due to the increasing global interest in natural health practices and personalized wellness solutions:

- Target Audience: The app is designed for individuals interested in holistic health, wellness, and Ayurveda. This includes health-conscious individuals, fitness enthusiasts, and those seeking natural remedies.
- Growing Interest in Ayurveda: With the global wellness industry seeing rapid growth, particularly in areas like diet, fitness, and mental well-being, an Ayurvedic health app aligns well with current market trends.
- Competitive Advantage: While there are other health apps in the market, few integrate Ayurvedic principles with a focus on personalized dosha-based recommendations. This unique approach can differentiate the app from competitors and attract a dedicated user base.

5. *Legal Feasibility*

Legal considerations are minimal but still important:

- Data Privacy: The app will need to comply with data privacy regulations, such as GDPR (for European users) and other regional data protection laws. User data must be securely stored, and transparent privacy policies must be in place.
- Health Disclaimer: Since the app provides health-related advice, it will be important to include a disclaimer stating that the app is not a substitute for professional medical advice and that users should consult a healthcare provider for serious conditions.

1.6Requirements of the Project

1. *Software Requirements*

Frontend:

- Development tools: React Native or Flutter (for cross-platform compatibility).

- Programming languages: JavaScript/TypeScript (React Native) or Dart (Flutter).
- UI/UX design tools: Figma, Sketch, or Adobe XD for designing the user interface.

Backend:

- Server-side languages: Node.js, Python (Flask/Django), or similar.
- Database: Firebase, MySQL, or PostgreSQL for storing user profiles, dosha assessments, and tracking data.
- Cloud hosting: AWS, Google Cloud, or Microsoft Azure for scalable storage and deployment. APIs:
- Use RESTful APIs for communication between the frontend and backend. ● Consider third-party APIs for features like push notifications (e.g., Firebase Cloud Messaging) or in-app purchases.

Version Control:

- Git for version control and collaboration among developers.

2. *Hardware*

Requirements Development Hardware

- Development machines (laptops/desktops) with sufficient specifications (8GB+ RAM, 256GB+ storage) for running development environments.
- Test devices (Android and iOS smartphones) to ensure compatibility and functionality across different platforms.

Server Infrastructure

- A cloud server with sufficient capacity for hosting the database, user data, and handling concurrent user requests.
- Load balancing and auto-scaling features for handling traffic spikes.

2. REQUIREMENTS

2.1Requirements

we have already said about requirements in the above context now we will know about the other requirements

2.2Non-Functional Requirements

These define the performance and usability criteria for the app:

1. *Usability*

- The app must be user-friendly and intuitive, ensuring that users can easily navigate the interface.
- Minimal steps should be required to access core functions such as dosha assessment and health

recommendations.

2. *Performance*

- The app should load quickly and handle multiple users simultaneously without lag. ○ Health tracking and recommendation generation should be processed efficiently. 3. Security:
- Secure user data through encryption to ensure the privacy of personal health information.
- Comply with data privacy laws, such as GDPR or regional privacy regulations, to safeguard user data.

3. *Scalability*

- The app must be able to scale to accommodate a growing user base, with the potential to handle thousands of users without compromising performance.

4. *Compatibility*

- The app should be compatible with both Android and iOS platforms.
- It should function across different screen sizes and mobile devices.

5. *Reliability*

- The app should function without crashing, ensuring high availability and reliability for users.

6. *Maintainability*

- The app's codebase should be easy to maintain, allowing for updates, bug fixes, and new features to be added over time.
- A version control system should be implemented to track changes in the code.

2.3Functional Requirements

These define the core functionalities the app must deliver:

1. *Dosha Assessment Module*

- A questionnaire-based system to assess the user's dominant dosha (Vata, Pitta, Kapha, or combination) based on their physical, emotional, and mental characteristics.
- Dynamic generation of results based on the answers provided by users.

2. *Personalized Health Recommendations*

- Provide dietary suggestions tailored to the user's dosha, highlighting foods to favor or ○ avoid.
- Suggest lifestyle practices (e.g., daily routines, exercises, meditation) that help balance the user's dosha.
- Offer seasonal recommendations and health tips based on Ayurvedic principles.

3. *Health Progress Tracking:*

- Allow users to track their health over time by logging symptoms or feelings.

- Update recommendations dynamically based on user inputs and progress.

4. *User Profile Management*

- Create user profiles to store personal details, dosha assessments, and health history.
- Enable users to edit and update their profile information and preferences.

5. *Educational Content Module*

- Provide educational material about Ayurveda, including detailed information on doshas, Ayurvedic diets, herbs, and seasonal wellness tips.
- Offer additional resources, such as links to Ayurvedic articles, books, or videos. 6. Notifications and Reminders:
- Implement a notification system that sends users regular reminders for recommended practices, such as meal suggestions, meditation, or exercise.

6. *Multi-language Support (optional for future development)*

- Enable support for multiple languages to reach a broader audience, especially in regions where Ayurveda is popular.

2.4User Interfaces

1. *Input Fields (Form Elements)*

- Hair Color:
  - Type: Dropdown (Select)
  - Options: Black, Brown, Blonde, Red
- Skin Color:
  - Type: Dropdown (Select)
  - Options: Light, Medium, Dark, Olive
- Skin Type:
  - Type: Dropdown (Select)
  - Options: Oily, Dry, Normal, Combination
- Nature:
  - Type: Dropdown (Select)
  - Options: Calm, Aggressive, Anxious, Balanced

2. *Buttons*

- Train and Predict Button:
  - Type: Button (Button)
  - Label: Train and Predict
  - Functionality: Triggers the neural network training and prediction.

- Start Over Button:
    - Type: Button (Button)
    - Label: Start Over
    - Functionality: Resets the input fields and prediction.

3. *Prediction Result Section*

- Predicted Dosha Display:
    - Type: Text (Result)
    - Displays the predicted Ayurvedic body type (Dosha) after training.
- Table of Dosha Information:
    - Type: Table (Table)
    - Columns:
    - Category: Describes what the row is about (e.g., "Recommended Foods to Eat"). 2. Details: Specific information based on the predicted Dosha, such as food
    - recommendations, foods to avoid, and suggested medicines.

4. *Additional Elements*

- Header:
    - Type: Heading (Header)
    - Label: Ayurvedic Body Type Classifier
- Form Container:
    - Type: Div (FormContainer)
    - Container for the input fields, holding them in a vertical layout.

2.5Constraints and Prerequisites Constraints

1. *Data Quality*

- The accuracy of dosha predictions is highly dependent on the quality of the input data provided by the user (hair color, skin type, nature, etc.).
- Incomplete or inaccurate input data could lead to incorrect dosha assessments and inappropriate recommendations.

2. *Limited Dataset*

- Ayurvedic body types (doshas) are complex and based on ancient knowledge, meaning your neural network's training data may be simplified.
- A small or unbalanced dataset could impact the model’s performance and prediction accuracy.

3. *User Variability*

- Ayurvedic dosha assessments are often subjective, and two users with similar physical attributes

might be classified differently based on small variations or other external factors.

- The app may not capture the full complexity of each user's unique constitution due to the simplicity of the questions.

4. *Real-time Performance*

- The model training might be resource-intensive and could cause delays on mobile devices with limited processing power.
- Depending on the device's performance, users may experience delays when the model is trained and predictions are made.

5. *Security and Privacy*

- Storing personal user data such as health status, preferences, and dosha types must be handled securely and in compliance with data privacy laws like GDPR.
- Sensitive health-related information needs encryption and strict access control. 6. Generalization of Recommendations:
- The dietary and lifestyle recommendations are general and based on Ayurvedic principles. These might not consider specific medical conditions or allergies, which can be a limitation for personalized health guidance.

6. *Cross-Platform Compatibility*

- Ensuring the app functions smoothly across different devices and operating systems (iOS, Android) is necessary but can be challenging due to variations in hardware capabilities and system versions.

2.6Prerequisites

1. *TensorFlow.js Knowledge*

- Basic understanding of machine learning principles and the ability to work with TensorFlow.js for creating and training the neural network model.

2. *React Development Skills*

- Familiarity with React.js, especially handling component states, props, and event handlers, is essential for developing the user interface.

3. *Ayurvedic Knowledge*

- A basic understanding of Ayurveda and dosha principles is required to accurately design the questionnaire and provide appropriate recommendations for each dosha type.

4. *JavaScript and Web Development*

- Proficiency in JavaScript, along with HTML and CSS, is needed to handle logic, design, and interactions within the app.

5. *Responsive Design*

- Knowledge of responsive web design principles to ensure the app's UI works well across different screen sizes and devices.

6. *Data Collection for Training*

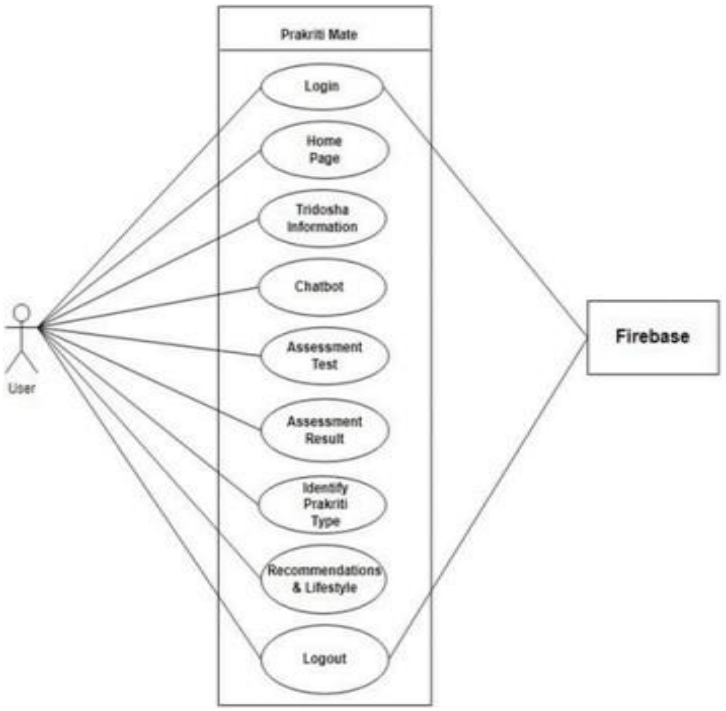
- Access to or creation of a dataset that contains dosha-related classifications for various combinations of hair color, skin type, and nature to train the model effectively. 7. Performance Optimization:
- Understanding how to optimize app performance for smooth functionality, including using web workers or offloading heavy processing tasks for slower devices.

7. *Security Best Practices*

- Implement secure data transmission and storage techniques, such as HTTPS for APIs, encryption for sensitive data, and adhering to security frameworks like OWASP.

3. ANALYSIS

3.1Use Case Model



A use case diagram visually represents the interactions between users (actors) and the system to show the functionalities available. Admin a secondary actor who manages the overall system and user activities.User The primary actor who interacts with the system to perform functions.

The user's Ayurvedic body type, or "Prakruti," is then determined by processing these inputs. After being recognized, PrakritiMate uses this analysis to deliver individualized health suggestions.

3.2Flow of Events

Questionnaire: A set of questions based on physical, mental, and emotional attributes.

- Physical (body frame, skin type, digestion, etc.).
- Mental (decision-making style, stress response, etc.).

- Emotional (mood swings, temper, etc.).

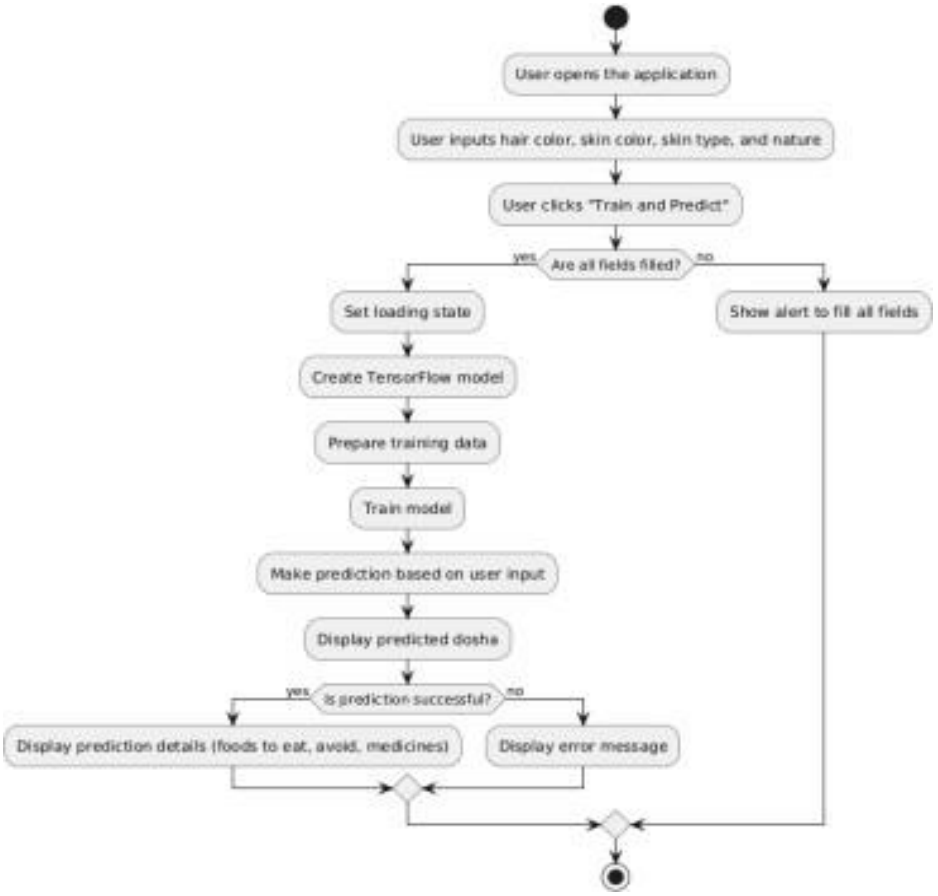
Result: Based on the answers, the app determines the user's Prakrithi (Vata, Pitta, Kapha, or a combination).

Explanation: Detailed description of the user's Prakrithi, including strengths, vulnerabilities, and tendencies.

Diet Plan: Customized food intake recommendations based on Prakrithi, season, and current imbalance (if any).

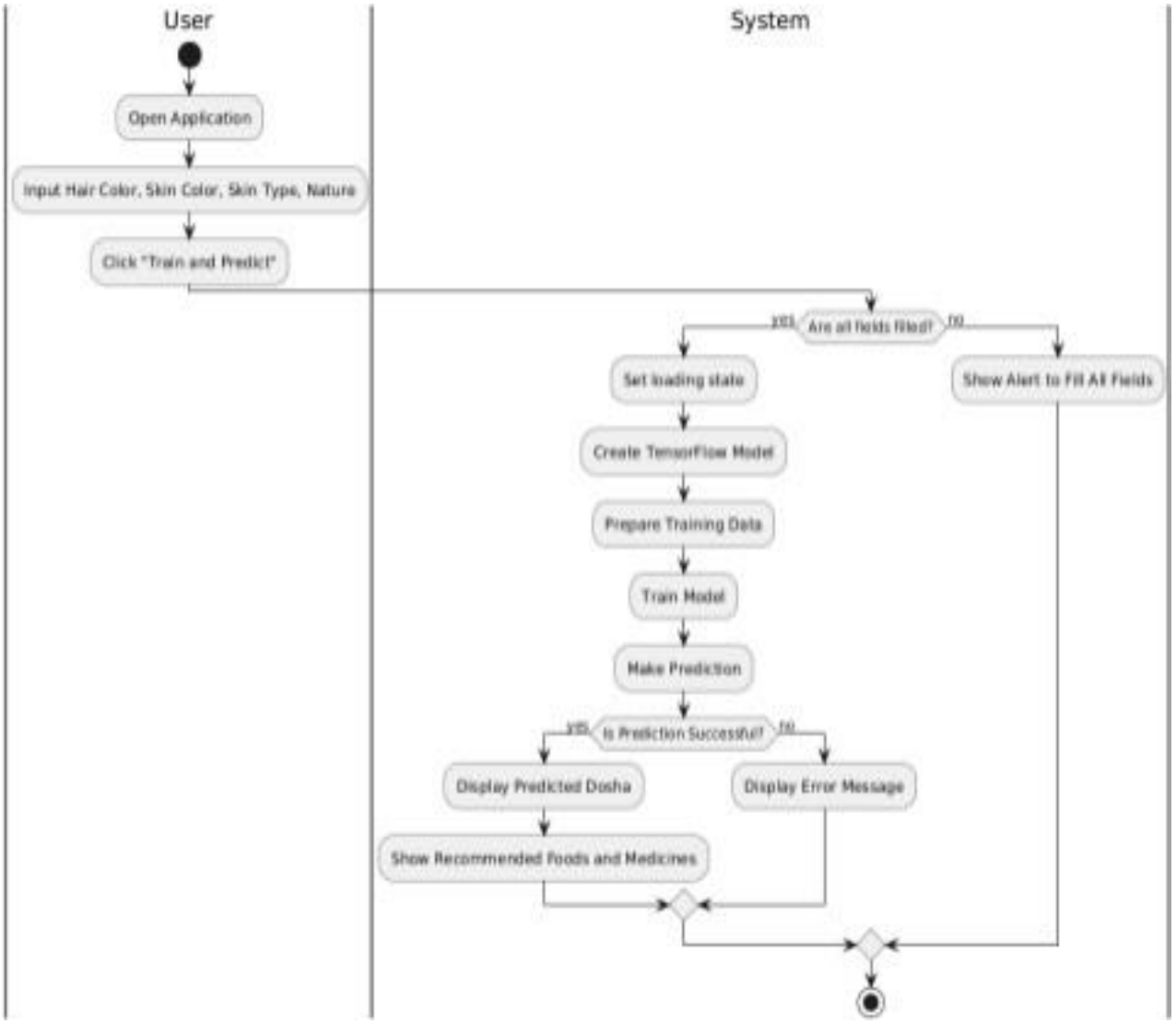
Lifestyle Tips: Daily routines (waking up, exercise, meditation) that align with the user's Prakrithi. Sleep Suggestions: Sleep patterns and suggestions for quality rest.

Mental Health Tips: Meditation, breathing techniques, or mindful activities tailored to balance Prakrithi.



3.3Activity Diagram

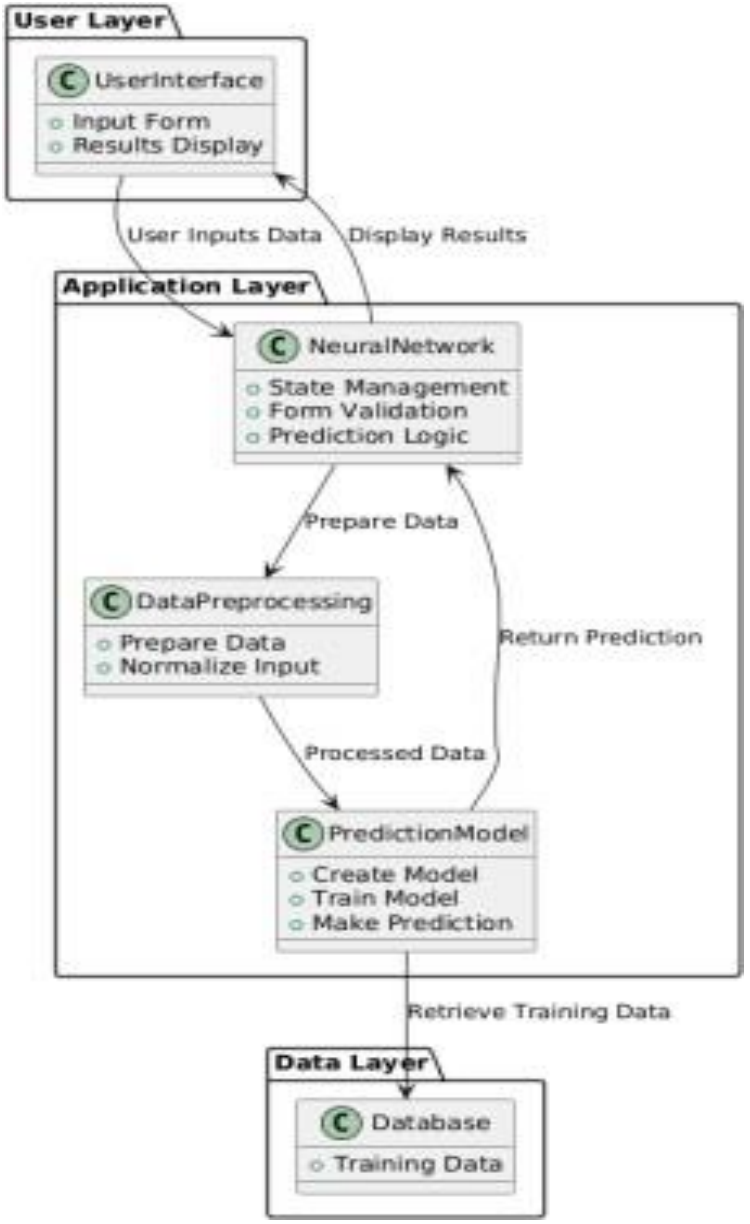
An activity diagram provides a visual representation of the flow of activities in the processes for ayurvedic app Below is a description of how to create an activity diagram based on the processes involved.



3.4System Architecture

Design
This architecture could be represented visually by showing:

- User interacts with the User Interface Layer (React app).
- User inputs are passed to the Processing Layer (TensorFlow.js), where the neural network is used for prediction.
- The Data Layer stores Dosha-related information and health suggestions.
- Predicted output is sent back to the UI and shown to the user.



The Ayurvedic body type classifier app is designed using a client-server architecture, with TensorFlow.js being used for on-device machine learning predictions. Below is an overview of the key components and their interactions:

1. *User Interface (UI) Layer*

- Frontend Framework: The UI is built with React.js for dynamic and responsive interactions.
- User Inputs: Users provide their health-related information, such as:
    - Hair Color
    - Skin Color
    - Skin Type
    - Nature (e.g., Calm, Aggressive)
- Predictions & Recommendations: After the model processes the input, the UI displays the predicted

Dosha type (Vata, Pitta, Kapha, etc.), recommended foods, and medicines.

2. *Processing Layer*

- TensorFlow.js Integration:
    - The machine learning model is built and trained using TensorFlow.js, allowing real-time predictions on the client side without the need for a backend server.
    - The model takes in user input as features (numerically encoded) and makes predictions on the user's body type (Dosha).
    - The model architecture is a feedforward neural network with layers as follows:
    - Input layer for features (4 input features: hair color, skin color, skin type, and nature)
- Hidden layers with ReLU activation for learning complex patterns
- Output layer with softmax activation to classify the user's Dosha type into 5 categories.

3. *Data Layer*

- User Data: The user’s input is converted into numerical format and fed into the model.
- Dosha Information: The app stores predefined Ayurvedic information related to Dosha types, such as:
    - Foods to eat and avoid
    - Recommended Ayurvedic medicines
    - Other health recommendations

4. *Model Training & Prediction*

- Training Data: The model is trained using a small synthetic dataset on the client side. In a real-world scenario, this could be done with a larger dataset to improve accuracy.
- On-Device Training: TensorFlow.js is used for training the model within the browser, ensuring the app works in offline or low-bandwidth environments.
- Prediction: After user inputs are collected, the trained model predicts the Dosha type in real-time and provides personalized health suggestions.

5. *Storage & Session Management*

- Local Storage (Optional): User preferences, inputs, and previous predictions can be stored locally for a smoother experience and to avoid redundant input.
- Session Management: Users can reset the app for a new prediction using the "Start Over" functionality.

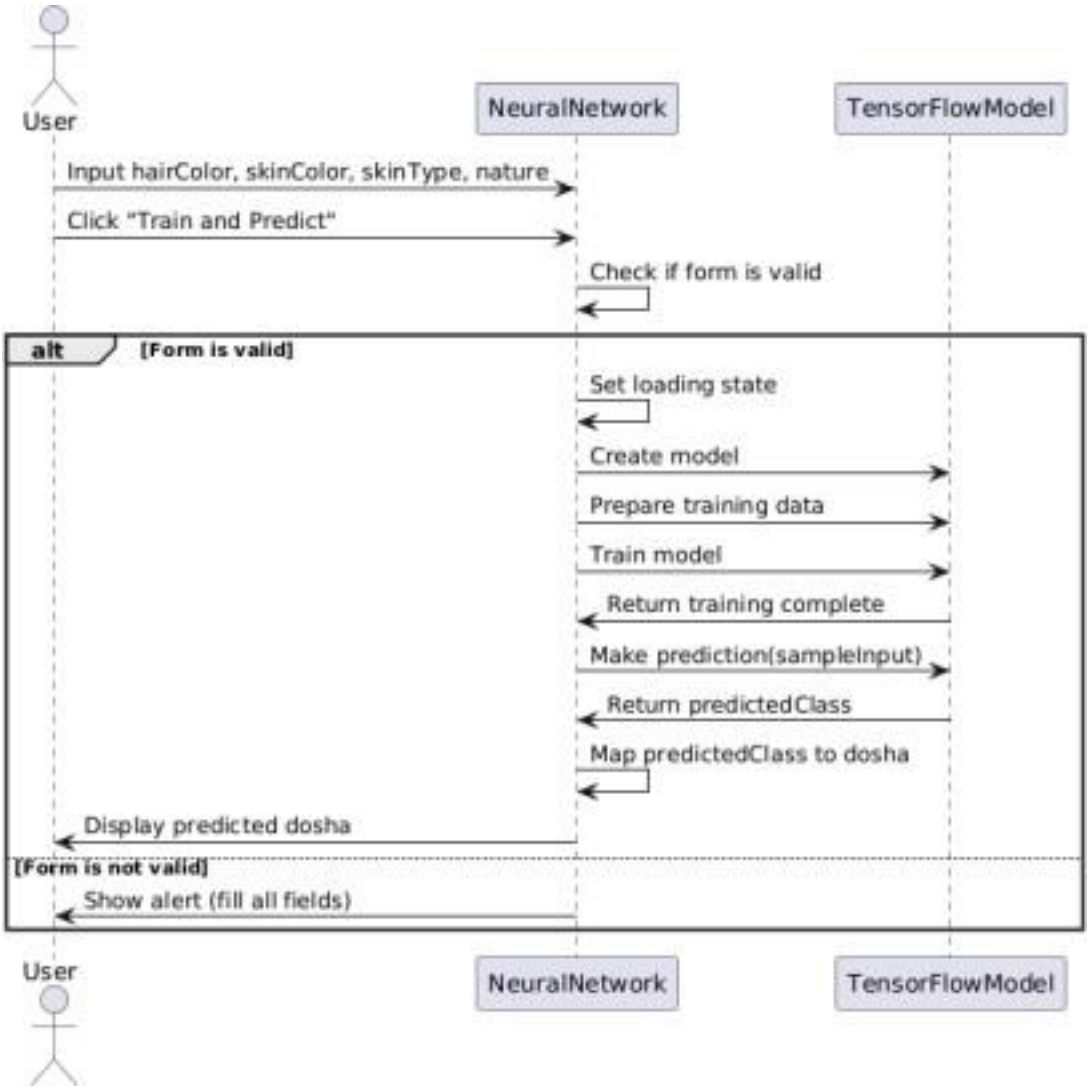
3.5Sequence Diagram

User Interface (UI): Acts as the medium through which users interact with the system, providing inputs and viewing results.

Processing Layer (TensorFlow.js): Responsible for executing the machine learning model and handling the prediction logic.

Data Layer: Stores static Ayurvedic data related to Dosha types, which the app uses to provide personalized recommendations.

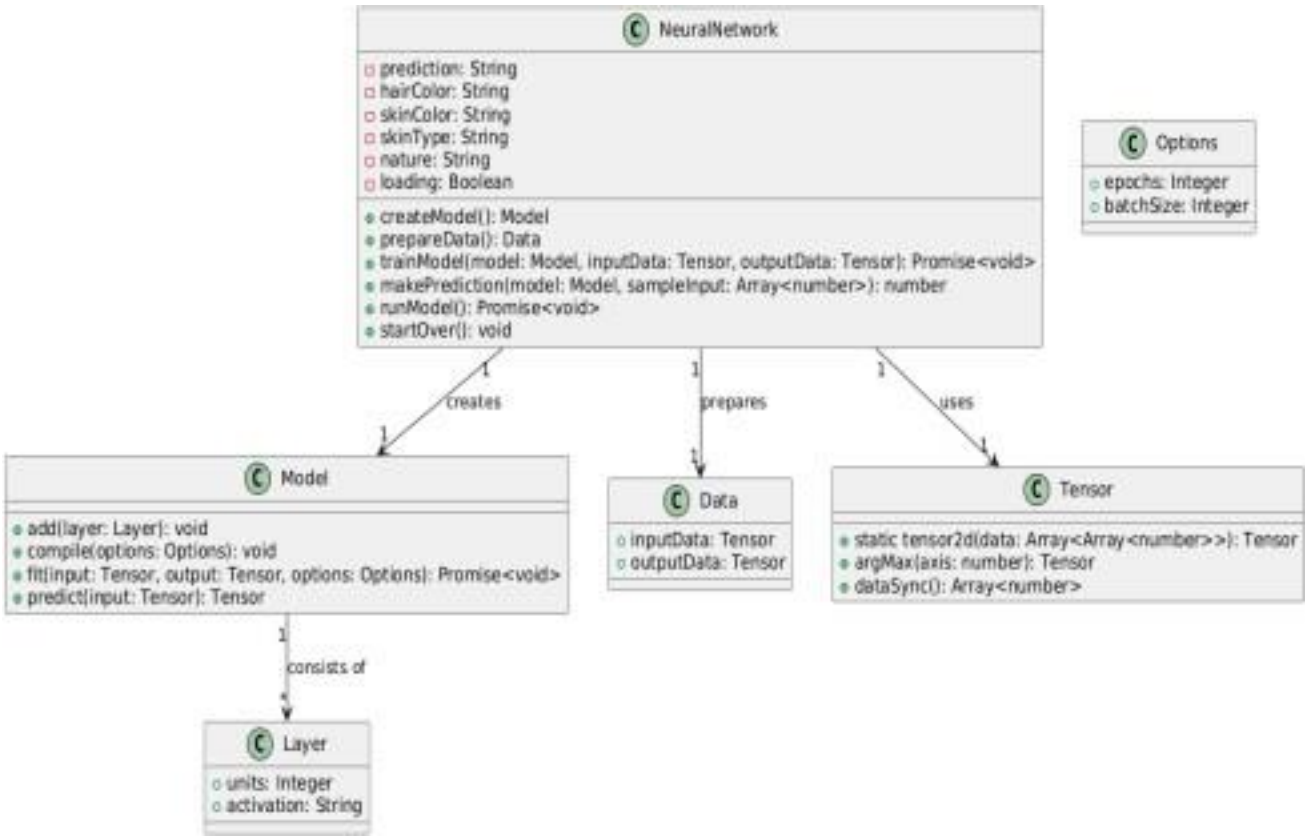
This sequence diagram emphasizes how the components interact, starting from user input, processing through TensorFlow.js, retrieving Ayurvedic data, and ultimately displaying the prediction results in a user-friendly manner. The seamless flow ensures a smooth user experience, allowing users to receive health recommendations based on their Dosha type.



3.6Class Diagram

- NeuralNetwork: The main component with properties and methods for managing the state and performing actions like training the model and making predictions.
- Model: Represents the TensorFlow model with methods for adding layers, compiling, fitting, and predicting.
- Data: Represents the input and output data for training the model.
- Tensor: Represents TensorFlow tensors, including methods for creating tensors and performing operations.
- Layer: Represents the layers of the neural network, including the number of units and activation functions.

- Options: Represents options for training the model, such as epochs and batch size.



4. IMPLEMENTATION

4.1Technologies Used

React

React is a popular JavaScript library for building user interfaces. It allows developers to create reusable UI components that manage their own state, making it easy to build complex user interfaces from small, isolated pieces of code.

Key Features:

- Component-based architecture: Encapsulates pieces of UI as components that can be reused and nested within other components.
- Virtual DOM: React uses a lightweight representation of the actual DOM, making updates fast by efficiently re-rendering only the parts of the UI that change.
- Hooks: Provides functions like useState and useEffect to manage component state and side effects without writing class components.
- JSX (JavaScript XML): A syntax extension that looks similar to HTML but allows you to write HTML-like code in JavaScript, making UI development more intuitive.
- Why React?: React is widely adopted due to its efficiency in handling complex state changes, ease of maintenance with components, and excellent ecosystem, including libraries for routing, state management, and testing.

Styled-components

Styled-components is a library for writing CSS directly within your JavaScript files. It leverages tagged template literals to style React components, allowing dynamic and scoped styles.

Key Features:

- CSS-in-JS: Styles are written in JavaScript, which improves developer productivity by co-locating styles with the components they affect.
- Scoped Styles: Styles are automatically scoped to the component, preventing CSS conflicts.
- Dynamic Styling: Allows you to pass props to components to dynamically change their styles.
- Theme Support: Helps you easily apply a consistent theme across the application using a ThemeProvider.
- Why Styled-components?: It improves maintainability by keeping component-specific styles close to their definition. Also, it reduces the risk of naming collisions in CSS by ensuring styles are scoped to the component.

TensorFlow.js

TensorFlow.js is a JavaScript library that brings machine learning (ML) capabilities to the browser or Node.js environments. It allows you to create, train, and deploy ML models in JavaScript, making machine learning more accessible for web developers.

Key Features:

- Create Neural Networks: Allows you to define neural networks with layers using
- tf.sequential() and train them with data.
- Train Models in the Browser: You can train models directly on the client-side, using
- the browser's computational resources (including WebGL for hardware acceleration). ○ Pre-trained Models: TensorFlow.js includes a number of pre-trained models for tasks like image recognition, object detection, and text processing.
- Tensor Operations: The library provides multi-dimensional arrays (tensors) and operations on them that are optimized for speed, utilizing hardware acceleration when possible.
- Why TensorFlow.js?: It enables machine learning tasks to be performed directly in the browser, which can reduce the need for server-side computation and provide real-time predictions for web apps.

JavaScript (ES6+)

JavaScript is the core programming language used in this application. ECMAScript 6 (ES6) or newer versions bring modern syntax and features that improve the language's usability and efficiency.

- Key Features in ES6+:
  - Arrow functions: Shorter syntax for writing functions (() => {}), making code cleaner.
  - Destructuring: Allows extracting values from objects or arrays into variables in a concise way.
  - Template literals: Multi-line strings and string interpolation using backticks.
  - Modules: Enables you to import and export functions, objects, or primitives between files,

improving code organization.

- Async/await: Modern syntax for handling asynchronous operations, making promises easier to work with.
- Promises: A pattern for dealing with asynchronous tasks, ensuring that JavaScript doesn't block other operations.

- Why JavaScript (ES6+)?: JavaScript is the foundation of web development, and modern ES6+ syntax significantly improves the readability and maintainability of the code, especially for larger applications.

HTML/CSS

Although you're not writing direct HTML or CSS in your React components, they are still foundational technologies for web development.

- HTML: Provides the structure of web pages. In React, JSX is an abstraction over HTML that allows you to write HTML-like syntax in JavaScript files.
- CSS: Used for styling web pages. With styled-components, CSS is integrated into JavaScript files to apply styles to React components. CSS controls the visual design and layout of your app.
- Why HTML/CSS?: HTML and CSS are the basic building blocks of any web application. Even though you're using JSX and styled-components, they ultimately compile down to HTML and CSS in the browser.

Browser-based Machine Learning

TensorFlow.js is a browser-friendly machine learning framework, and by using it in React, you're performing machine learning tasks directly in the browser:

Advantages of browser-based ML:

- Client-side Computation: Offloads computational tasks to the user's device, reducing the load on the server.
- Real-time Processing: Predictions and model inference happen in real-time within the browser, enabling faster and more interactive experiences.
- No Need for Backend Infrastructure: In some cases, you can avoid the need for server-side APIs for predictions or model inference.
- Data Privacy: Since data never leaves the client, it enhances privacy, especially for sensitive user data.

Summary of How These Technologies Work Together:

- React provides the core structure and state management for the app.
- Styled-components make the app visually appealing by applying CSS-in-JS to ensure the UI is clean, responsive, and easy to maintain.
- TensorFlow.js enables real-time machine learning within the browser, allowing you to create and train a neural network directly in the web app.
- JavaScript (ES6+) powers the logic behind user input, neural network setup, and predictions. ● HTML/CSS are the foundational web technologies, with React's JSX and styled-components handling their roles in a more dynamic and scoped manner.

This combination makes your app powerful, enabling both user-friendly interfaces and cutting-edge machine learning without needing external servers for computation.

4.2Code Implementation

Login page Login.jsx

```
import React, { useState } from "react";
import "./app.css"; // Ensure the CSS is included


const Login = ({ onLogin }) => {
const [username, setUsername] = useState(""); const [password, setPassword] = useState("");
const [error, setError] = useState("");
const handleSubmit = (e) => { e.preventDefault();

// For simplicity, a basic hardcoded login check
if (username === "admin" && password === "password") { onLogin(true); // Call the onLogin
function passed from App.jsx
} else {
setError("Invalid username or password");
}
};


return (
<div className="login-container">
<h2 className="login-header">Login</h2>
<form onSubmit={handleSubmit} className="login-form">
<div className="input-group">
<label className="label">Username</label>
<input type="text"
className="input" value={username}
onChange={(e) => setUsername(e.target.value)} placeholder="Enter your username"
required
/>
</div>
<div className="input-group">
<label className="label">Password</label>
<input type="password" className="input" value={password}
onChange={(e) => setPassword(e.target.value)} placeholder="Enter your password"
required
/>
</div>
{error && <p className="error">{error}</p>}
<button type="submit" className="login-button"> Login
</button>
</form>
</div>
);
```

```
};
export default Login
```

Neural.jsx

```
import React, { useState } from 'react';
import styled from
'styled-components'; import * as tf from '@tensorflow/tfjs';


const Container = styled.div` max-width: 600px;
margin: auto; padding: 20px; text-align: center;
background-color: #e3f2fd; border-radius: 10px;
box-shadow: 0 4px 8px rgba(0, 0, 0,
0.1); `;

const Header = styled.h2` color: #333;
margin-bottom: 20px;
`;

const FormContainer = styled.div` display: flex;
flex-direction: column; align-items: flex-start;
`;

const Label = styled.label` margin: 10px 0;
font-size: 16px;
`;

const Select = styled.select` margin-left: 10px; padding: 5px;
border-radius: 5px; border: 1px solid #ccc;
transition: border-color 0.3s;

&:hover {
border-color: #4caf50;
}
`;

const Button = styled.button` margin-top: 20px;
padding: 10px 15px; font-size: 16px;
background-color: #4caf50; color: white;
border: none; border-radius: 5px; cursor: pointer;
transition: background-color 0.3s, transform 0.2s;

&:hover {
background-color: #45a049; transform: scale(1.05);
}
```

```
&:active {
transform: scale(0.95);
}
`;

const Result = styled.div` margin-top: 20px;
font-size: 18px; font-weight: bold; color: #4caf50;
`;

const Table = styled.table` width: 100%;
margin-top: 10px;
border-collapse: collapse;

th, td {
border: 1px solid #ddd; padding: 8px;
text-align: left;
}

th {
background-color: #4caf50; color: white;
}

tr:nth-child(even) { background-color: #f2f2f2;
}

tr:hover {
background-color: #ddd;
}
`;

const DoshaList = styled.div` text-align: left;
margin-top: 10px;
`;


const NeuralNetwork = () => {
const [prediction, setPrediction] = useState(null); const [hairColor, setHairColor] = useState('');
const [skinColor, setSkinColor] = useState(''); const [skinType, setSkinType] = useState(''); const
[nature, setNature] = useState('');
const createModel = () => { const model = tf.sequential();
model.add(tf.layers.dense({ units: 8, inputShape: [4], activation: 'relu' }));
model.add(tf.layers.dense({ units: 12, activation: 'relu' })); model.add(tf.layers.dense({ units: 8,
activation: 'relu' })); model.add(tf.layers.dense({ units: 5, activation: 'softmax' }));
model.compile({ optimizer: 'adam', loss: 'categoricalCrossentropy', metrics: ['accuracy'] }); return
model;
};


const prepareData = () => { const inputData = tf.tensor2d([
```

```
[0, 1, 0, 2],
[2, 0, 1, 1],
[1, 2, 0, 0],
[1, 1, 2, 1]
]);

const outputData = tf.tensor2d([ [1, 0, 0, 0, 0],
[0, 1, 0, 0, 0],
[0, 0, 1, 0, 0],
[0, 0, 0, 1, 0]
]);

return { inputData, outputData };
};

const trainModel = async (model, inputData, outputData) => { return await model.fit(inputData,
outputData, {
epochs: 50,
batchSize: 4
});
};

const makePrediction = (model, sampleInput) => { const inputTensor =
tf.tensor2d([sampleInput]); const prediction = model.predict(inputTensor);
const predictedClass = prediction.argMax(1).dataSync()[0]; return predictedClass;
};


const runModel = async () => {
const model = createModel();
const { inputData, outputData } = prepareData();


await trainModel(model, inputData, outputData);


const sampleInput = [
hairColor === 'Black' ? 0 : hairColor === 'Brown' ? 1 : hairColor === 'Blonde' ? 2 : 3, skinColor
=== 'Light' ? 0 : skinColor === 'Medium' ? 1 : skinColor === 'Dark' ? 2 : 3, skinType === 'Oily' ?
0 : skinType === 'Dry' ? 1 : skinType === 'Normal' ? 2 : 3, nature === 'Calm' ? 0 : nature ===
'Aggressive' ? 1 : nature === 'Anxious' ? 2 : 3 ];


const predictedClass = makePrediction(model, sampleInput);


const bodyTypes = ['Vata', 'Pitta', 'Kapha', 'Vata-Pitta', 'Pitta-Kapha'];

const predictedDosha = bodyTypes[predictedClass]; setPrediction(predictedDosha);
```

```
};


const startOver = () => { setPrediction(null); setHairColor(''); setSkinColor(''); setSkinType('');
setNature('');
};

const doshaInfo = { Vata: {
eat: 'warm, “moist”, and soft foods (e.g., berries, bananas, peaches, cooked vegetables, oats,
brown rice, lean meat, eggs, dairy)',
avoid: 'bitter, dried, and cold foods (e.g., raw vegetables, cold desserts, dried fruit, nuts, seeds)',
medicine: 'Ashwagandha, Ghee, and Sweeteners like Honey'
},
Pitta: {
eat: 'light, cold, sweet, and energizing foods (e.g., fruits, non-starchy vegetables, oats, eggs)',
avoid: 'heavy, spicy, and sour foods (e.g., red meat, potatoes, hot spices)', medicine:
'Chyawanprash, Aloe Vera, and Cooling herbs like Mint'
},
Kapha: {
eat: 'spicy, acidic, and filling foods (e.g., most fruits and vegetables, whole grains, eggs, low-fat
cheese, unprocessed meats, hot spices)',
avoid: 'heavy, “fatty” foods (e.g., fats, oils, processed foods, nuts, seeds)', medicine: 'Ginger,
Turmeric, and stimulating herbs like Black Pepper'
},
};

return (
<Container>
<Header>Ayurvedic Body Type Classifier</Header>
<FormContainer>
<Label> Hair Color:
<Select value={hairColor} onChange={(e) => setHairColor(e.target.value)}>
<option value="">Select</option>
<option value="Black">Black</option>
<option value="Brown">Brown</option>
<option value="Blonde">Blonde</option>
<option value="Red">Red</option>
</Select>
</Label>
<Label>
<Label></Label> Skin Color:
<Select value={skinColor} onChange={(e) => setSkinColor(e.target.value)}>
<option value="">Select</option>
<option value="Light">Light</option>
<option value="Medium">Medium</option>
<option value="Dark">Dark</option>
<option value="Olive">Olive</option>
</Select>
</Label>
```

```
<Label> Skin Type:
<Select value={skinType} onChange={(e) => setSkinType(e.target.value)}>
<option value="">Select</option>
<option value="Oily">Oily</option>
<option value="Dry">Dry</option>
<option value="Normal">Normal</option>
<option value="Combination">Combination</option>
</Select>
</Label>
<Label> Nature:
<Select value={nature} onChange={(e) => setNature(e.target.value)}>
<option value="">Select</option>
<option value="Calm">Calm</option>
<option value="Aggressive">Aggressive</option>
<option value="Anxious">Anxious</option>
<option value="Balanced">Balanced</option>
</Select>
</Label>
<Button onClick={runModel}>Train and Predict</Button>
<Button onClick={startOver} style={{ backgroundColor: '#f44336', marginTop: '10px' }}> Start
Over
</Button>
</FormContainer>

{prediction && (
<Result>
<h3>Your Predicted Dosha: {prediction}</h3>
<Table>
<thead>
<tr>
<th>Category</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recommended Foods to Eat:</td>
<td>{doshaInfo[prediction].eat}</td>
</tr>
<tr>
<td>Foods to Avoid:</td>
<td>{doshaInfo[prediction].avoid}</td>
</tr>
<tr>
<td>Suggested Medicines:</td>
<td>{doshaInfo[prediction].medicine}</td>
</tr>
</tbody>
</Table>
```

```
</Result>
)}
</Container>
);
};

export default NeuralNetwork;
```

Launch.json

```
{
// Use IntelliSense to learn about possible attributes.
// Hover to view descriptions of existing attributes.
// For more information, visit: https://go.microsoft.com/fwlink/?linkid=830387 "version": "0.2.0",
"configurations": [
{
"type": "chrome",
"request": "launch",
"name": "Launch Chrome against localhost", "url": "http://localhost:8080",
"webRoot": "${workspaceFolder}"
}
]
}
```

App.jsx

```
import React, { useState } from "react"; import "./app.css";
import Login from "./Login";
import Neural from "./Neural"; // Assuming Neural is your main classifier component const


App = () => {


const [isAuthenticated, setIsAuthenticated] = useState(false);


const handleLogin = (status) => { setIsAuthenticated(status);
};


return (
<div>
{!isAuthenticated ? (
<Login onLogin={handleLogin} />
) : (
<Neural />
)}
</div>
);
};
```

App.css

```
#root {
max-width: 1280px; margin: 0 auto; padding: 2rem;
text-align: center;
}


.logo height: 6em;
padding: 1.5em; will-change: filter;
transition: filter 300ms;
}
.logo:hover {
filter: drop-shadow(0 0 2em #646cffaa); }
.logo.react:hover {
filter: drop-shadow(0 0 2em #61dafbaa);
}



@keyframes logo-spin { from {
transform: rotate(0deg);
}
to {
transform: rotate(360deg);
}
}


@media (prefers-reduced-motion: no-preference) { a:nth-of-type(2) .logo {
animation: logo-spin infinite 20s linear;
}

}


.card { padding: 2em;
}


.read-the-docs { color: #888;
}
/* Login Page Styles */
.login-container { max-width: 400px; margin: 50px auto; padding: 20px;
background-color: #f0f0f0; border-radius: 10px;
box-shadow: 0 4px 8px rgba(0, 0, 0,
0.1); text-align: center;

}
30
```

```
.login-header { font-size: 24px;
margin-bottom: 20px; color: #333;
}
.input-group {
margin-bottom: 15px; text-align: left;
}
.label {
font-size: 14px; color: #333;
}
.input { width: 100%;
padding: 10px;
border: 1px solid #ccc; border-radius: 5px; margin-top: 5px;
}
.login-button { width: 100%; padding: 10px;
background-color: #4caf50; color: white;
border: none; border-radius: 5px; cursor: pointer; font-size: 16px;
}
.login-button:hover { background-color: #45a049;
}
.error { color: red;
font-size: 14px;
margin-top: 10px;
}
```

5. USER SCREENS

5.1Login screen

The login page in your application provides a simple and secure way for users to log in by entering their credentials. It consists of several key components that work together to handle user input, validate credentials, and display feedback if the login attempt is unsuccessful.

The overall goal of this login page is to provide a seamless and efficient login process while ensuring that users receive clear feedback during their interaction.

5.2Users Interface Screen

The page features a clean, user-friendly form where users can select their characteristics from dropdown menus.Hair Color, Skin Color, Skin Type, and Nature are presented as selection options, allowing users to provide input relevant to their Ayurvedic assessment.

Upon submission of the form, the user's selections are processed through a trained neural network model that predicts their Ayurvedic dosha. The dosha prediction is displayed prominently, providing immediate feedback to the user.

5.3Output

The output of the Ayurvedic Body Type Classifier program will vary based on the user's input selections for hair color, skin color, skin type, and nature. Below is a description of the expected output after the user interacts with the application

The output effectively summarizes the user's Ayurvedic body type along with actionable dietary and medicinal advice, enhancing the user's understanding and application of Ayurvedic principles in their lifestyle.

6. CONCLUSIONS AND FUTURE WORK

6.1 Conclusions

The Ayurvedic Body Type Classifier project successfully integrates modern web technologies with traditional Ayurvedic principles, providing users with a unique and interactive way to understand their body types (doshas) based on specific personal characteristics.

Key Takeaways

1. Educational Tool

The application serves as an educational resource, introducing users to the concepts of Ayurveda and helping them gain insights into their health and wellness based on their individual traits.

2. Machine Learning Integration

By utilizing TensorFlow.js, the project demonstrates the application of machine learning in real-time classification tasks. Users can experience how AI can predict their dosha based on a simple input model, enhancing their understanding of both technology and Ayurvedic practices.

3. User-Friendly Interface

The well-designed user interface promotes engagement and accessibility. With clearly labeled input fields and informative output tables, users can easily navigate the application and comprehend their results.

4. Practical Applications

The recommendations provided by the application (dietary suggestions and herbal remedies) encourage users to apply Ayurvedic principles in their daily lives. This can lead to improved health outcomes and a greater awareness of personal well-being.

5. Opportunities for Improvement

While the current model provides a basic classification based on a limited dataset, future enhancements could include:

- More Comprehensive Training Data: Incorporating a wider variety of input examples to improve prediction accuracy.
- User Authentication: Adding user accounts to save personal results and track changes over time.
- Mobile Optimization: Ensuring the application is fully responsive for a better experience on mobile devices.

6. *Cultural Significance*

By merging traditional Ayurvedic concepts with contemporary technology, the project not only promotes health and wellness but also fosters an appreciation for holistic healing practices.

Final Thoughts

The Ayurvedic Body Type Classifier project exemplifies the convergence of technology and holistic health. It empowers users to make informed decisions about their health based on personalized insights, thereby bridging the gap between ancient wisdom and modern science. Moving forward, the project holds the potential to expand its capabilities, further enriching the user experience and the application of Ayurvedic practices in everyday life.

6.2 Future Enhancements

To further improve the Ayurvedic Body Type Classifier and enhance user experience, several future enhancements can be implemented. These improvements can expand the functionality, increase accuracy, and foster a more engaging interaction with users. Below are some suggested enhancements:

1. *Expanded Dataset and Model Training*

Diverse Inputs: Collect a more comprehensive and varied dataset that includes a wider range of hair colors, skin tones, and personality traits to improve the model's accuracy.

Advanced Algorithms: Explore more complex machine learning algorithms or models, such as decision trees or ensemble methods, to improve classification accuracy.

Continuous Learning: Implement a mechanism for the model to learn from user feedback, allowing it to improve over time based on real-world inputs.

2. *User Personalization*

User Accounts: Create user accounts to enable users to save their profiles, track changes in their dosha over time, and receive personalized recommendations.

Tailored Recommendations: Provide customized dietary and lifestyle suggestions based on individual user preferences, allergies, and health conditions.

3. *Enhanced User Interface and Experience*

Responsive Design: Optimize the application for mobile devices to ensure a seamless user experience across different platforms.

Interactive Features: Incorporate visual aids, such as infographics or videos, to educate users about Ayurvedic principles and their dosha types.

User Feedback: Implement feedback mechanisms where users can rate the accuracy of predictions and the usefulness of the recommendations, helping to refine the system.

4. *Integration of Additional Ayurvedic Concepts*

Seasonal Recommendations: Include suggestions that vary by season, allowing users to adjust their diets and lifestyles according to Ayurvedic principles related to seasonal changes.

Ayurvedic Remedies: Expand the information provided to include various Ayurvedic remedies, treatments, and practices tailored to each dosha.

5. *Machine Learning Model Optimization*

Hyperparameter Tuning: Experiment with different hyperparameters during model training to find the optimal configuration that yields better accuracy.

Cross-Validation: Implement cross-validation techniques to assess the model’s performance more rigorously and prevent overfitting.

6. *Community and Social Features*

Forums and Discussion Boards: Create a community platform where users can share experiences, recipes, and tips related to their dosha.

Social Media Integration: Allow users to share their results and recommendations on social media, encouraging others to explore the application.

7. *Multilingual Support*

Language Options: Offer the application in multiple languages to reach a broader audience and cater to users from diverse linguistic backgrounds.

8. *Integration with Health Tracking Apps*

API Connections: Develop APIs to integrate with health tracking applications, allowing users to sync their Ayurvedic insights with other health data they may be tracking.

Implementing these future enhancements will not only improve the functionality and accuracy of the Ayurvedic Body Type Classifier but also create a more engaging and personalized experience for users. By continually adapting and expanding the application, it can serve as a valuable tool for promoting health and well-being through the principles of Ayurveda in a modern context.

REFERENCES

1. Murthy, G., and R. Shankar. "Composite Fermions." (1998): 254-306.
2. Mahalakshmi, A., Goud, N. S., & Murthy, G. V. (2018). A survey on phishing and it's detection techniques based on support vector method (Svm) and software defined networking (sdn). *International Journal of Engineering and Advanced Technology*, *8*(2), 498-503.
3. Murthy, G., & Shankar, R. (2002). Semiconductors II-Surfaces, interfaces, microstructures, and related topics-Hamiltonian theory of the fractional quantum Hall effect: Effect of Landau level mixing. *Physical Review-Section B-Condensed Matter*, *65*(24), 245309-245309.
4. Murthy, G. V. K., Sivanagaraju, S., Satyanarayana, S., & Rao, B. H. (2014). Optimal placement of DG in distribution system to mitigate power quality disturbances. *International Journal of Electrical and Computer Engineering*, *7*(2), 266-271.
5. Muraleedharan, K., Raghavan, R., Murthy, G. V. K., Murthy, V. S. S., Swamy, K. G., & Prasanna, T. (1989). An investigation on the outbreaks of pox in buffaloes in Karnataka.
6. Murthy, G. V. K., Sivanagaraju, S., Satyanarayana, S., & Rao, B. H. (2012). Reliability improvement of radial distribution system with distributed generation. *International Journal of Engineering Science and Technology (IJEST)*, *4*(09), 4003-4011.
7. Gowda, B. M. V., Murthy, G. V. K., Upadhye, A. S., & Raghavan, R. (1996). Serotypes of Escherichia coli from pathological conditions in poultry and their antibiogram.
8. Balasubbareddy, M., Murthy, G. V. K., & Kumar, K. S. (2021). Performance evaluation of different structures of power system stabilizers. *International Journal of Electrical and Computer Engineering (IJECE)*, *11*(1), 114-123.
9. Murthy, G. V. K., & Sivanagaraju, S. (2012). S. Satyana rayana, B. Hanumantha Rao," Voltage stability index of radial distribution networks with distributed generation,". *Int. J. Electr. Eng*, *5*(6), 791-803.
10. Anuja, P. S., Kiran, V. U., Kalavathi, C., Murthy, G. N., & Kumari, G. S. (2015). Design of elliptical patch antenna with single & double U-slot for wireless applications: a comparative approach. *International Journal of Computer Science and Network Security (IJCSNS)*, *15*(2), 60.
11. Siva Prasad, B. V. V., Mandapati, S., Kumar Ramasamy, L., Boddu, R., Reddy, P., & Suresh Kumar, B. (2023). Ensemble-based cryptography for soldiers' health monitoring using mobile ad hoc networks. *Automatika: časopis za automatiku, mjerenje, elektroniku, računarstvo i komunikacije*, *64*(3), 658-671.
12. Siva Prasad, B. V. V., Sucharitha, G., Venkatesan, K. G. S., Patnala, T. R., Murari, T., & Karanam, S. R. (2022). Optimisation of the execution time using hadoop-based parallel machine learning on computing clusters. In *Computer Networks, Big Data and IoT: Proceedings of ICCBI 2021* (pp. 233-244). Singapore: Springer Nature Singapore.

13. Prasad, B. V., & Ali, S. S. (2017). Software–defined networking based secure rout-ing in mobile ad hoc network. *International Journal of Engineering & Technology*, *7*(1.2), 229.
14. Elechi, P., & Onu, K. E. (2022). Unmanned Aerial Vehicle Cellular Communication Operating in Non-terrestrial Networks. In *Unmanned Aerial Vehicle Cellular Communications* (pp. 225-251). Cham: Springer International Publishing.
15. Prasad, B. V. V. S., Mandapati, S., Haritha, B., & Begum, M. J. (2020, August). Enhanced Security for the authentication of Digital Signature from the key generated by the CSTRNG method. In *2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT)* (pp. 1088-1093). IEEE.
16. Alapati, N., Prasad, B. V. V. S., Sharma, A., Kumari, G. R. P., Veeneetha, S. V., Srivalli, N., ... & Sahitya, D. (2022, November). Prediction of Flight-fare using machine learning. In *2022 International Conference on Fourth Industrial Revolution Based Technology and Practices (ICFIRTP)* (pp. 134-138). IEEE.
17. Alapati, N., Prasad, B. V. V. S., Sharma, A., Kumari, G. R. P., Bhargavi, P. J., Alekhya, A., ... & Nandini, K. (2022, November). Cardiovascular Disease Prediction using machine learning. In *2022 International Conference on Fourth Industrial Revolution Based Technology and Practices (ICFIRTP)* (pp. 60-66). IEEE.
18. Mukiri, R. R., Kumar, B. S., & Prasad, B. V. V. (2019, February). Effective Data Collaborative Strain Using RecTree Algorithm. In *Proceedings of International Conference on Sustainable Computing in Science, Technology and Management (SUSCOM), Amity University Rajasthan, Jaipur-India*.
19. Rao, B. T., Prasad, B. V. V. S., & Peram, S. R. (2019). Elegant Energy Competent Lighting in Green Buildings Based on Energetic Power Control Using IoT Design. In *Smart Intelligent Computing and Applications: Proceedings of the Second International Conference on SCI 2018, Volume 1* (pp. 247-257). Springer Singapore.
20. Someswar, G. M., & Prasad, B. V. V. S. (2017, October). USVGM protocol with two layer architecture for efficient network management in MANET'S. In *2017 2nd International Conference on Communication and Electronics Systems (ICCES)* (pp. 738-741). IEEE.
21. Hnamte, V., & Balram, G. (2022). Implementation of Naive Bayes Classifier for Reducing DDoS Attacks in IoT Networks. *Journal of Algebraic Statistics*, *13*(2), 2749-2757.
22. Balram, G., Poornachandrarao, N., Ganesh, D., Nagesh, B., Basi, R. A., & Kumar, M. S. (2024, September). Application of Machine Learning Techniques for Heavy Rainfall Prediction using Satellite Data. In *2024 5th International Conference on Smart Electronics and Communication (ICOSEC)* (pp. 1081-1087). IEEE.
23. Subrahmanyam, V., Sagar, M., Balram, G., Ramana, J. V., Tejaswi, S., & Mohammad, H. P. (2024, May). An Efficient Reliable Data Communication For Unmanned Air Vehicles (UAV) Enabled Industry Internet of Things (IIoT). In *2024 3rd International Conference on Artificial Intelligence For Internet of Things (AIIoT)* (pp. 1-4). IEEE.
24. KATIKA, R., & BALRAM, G. (2013). Video Multicasting Framework for Extended Wireless Mesh Networks Environment. *pp-427-434*, *IJSRET*, *2*(7).
25. Prasad, P. S., & Rao, S. K. M. (2017). HIASA: Hybrid improved artificial bee colony and simulated annealing based attack detection algorithm in mobile ad-hoc networks (MANETs). *Bonfring International Journal of Industrial Engineering and Management Science*, *7*(2), 01-12.
26. Prasad, P. S., & Rao, S. K. M. (2017). A Survey on Performance Analysis of ManetsUnder Security Attacks. *network*, *6*(7).
27. Reddy, P. R. S., & Ravindranath, K. (2024). Enhancing Secure and Reliable Data Transfer through Robust Integrity. *Journal of Electrical Systems*, *20*(1s), 900-910.
28. REDDY, P. R. S., & RAVINDRANATH, K. (2022). A HYBRID VERIFIED RE-ENCRYPTION INVOLVED PROXY SERVER TO ORGANIZE THE GROUP DYNAMICS: SHARING AND REVOCATION. *Journal of Theoretical and Applied Information Technology*, *100*(13).
29. Reddy, P. R. S., Ram, V. S. S., Greshma, V., & Kumar, K. S. Prediction of Heart Healthiness.
30. Reddy, P. R. S., Reddy, A. M., & Ujwala, B. IDENTITY PRESERVING IN DYNAMIC GROUPS FOR DATA SHARING AND AUDITING IN CLOUD.
31. Madhuri, K., Viswanath, N. K., & Gayatri, P. U. (2016, November). Performance evaluation of AODV under Black hole attack in MANET using NS2. In *2016 international conference on ICT in Business Industry & Government (ICTBIG)* (pp. 1-3). IEEE.
32. Kovoor, M., Durairaj, M., Karyakarte, M. S., Hussain, M. Z., Ashraf, M., & Maguluri, L. P. (2024). Sensor-enhanced wearables and automated analytics for injury prevention in sports. *Measurement: Sensors*, *32*, 101054.
33. Rao, N. R., Kovoor, M., Kishor Kumar, G. N., & Parameswari, D. V. L. (2023). Security and privacy in smart farming: challenges and opportunities. *International Journal on Recent and Innovation Trends in Computing and Communication*, *11*(7 S).

34. Madhuri, K. (2023). Security Threats and Detection Mechanisms in Machine Learning. *Handbook of Artificial Intelligence*, 255.
35. DASTAGIRAIAH, D. (2024). A SYSTEM FOR ANALYSING CALL DROP DYNAMICS IN THE TELECOM INDUSTRY USING MACHINE LEARNING AND FEATURE SELECTION. *Journal of Theoretical and Applied Information Technology*, *102*(22).
36. Sukhavasi, V., Kulkarni, S., Raghavendran, V., Dastagiraiah, C., Apat, S. K., & Reddy, P. C. S. (2024). Malignancy Detection in Lung and Colon Histopathology Images by Transfer Learning with Class Selective Image Processing.
37. Sudhakar, R. V., Dastagiraiah, C., Pattem, S., & Bhukya, S. (2024). Multi-Objective Reinforcement Learning Based Algorithm for Dynamic Workflow Scheduling in Cloud Computing. *Indonesian Journal of Electrical Engineering and Informatics (IJEEI)*, *12*(3), 640-649.
38. PushpaRani, K., Roja, G., Anusha, R., Dastagiraiah, C., Srilatha, B., & Manjusha, B. (2024, June). Geological Information Extraction from Satellite Imagery Using Deep Learning. In *2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT)* (pp. 1-7). IEEE.
39. Sravan, K., Rao, L. G., Ramineni, K., Rachapalli, A., & Mohmmad, S. (2024). Analyze the Quality of Wine Based on Machine Learning Approach Check for updates. *Data Science and Applications: Proceedings of ICDSA 2023, Volume 3*, *820*, 351.
40. Chandhar, K., Ramineni, K., Ramakrishna, E., Ramana, T. V., Sandeep, A., & Kalyan, K. (2023, December). Enhancing Crop Yield Prediction in India: A Comparative Analysis of Machine Learning Models. In *2023 3rd International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON)* (pp. 1-4). IEEE.
41. Ramineni, K., Shankar, K., Shabana, Mahender, A., & Mohmmad, S. (2023, June). Detecting of Tree Cutting Sound in the Forest by Machine Learning Intelligence. In *International Conference on Power Engineering and Intelligent Systems (PEIS)* (pp. 303-314). Singapore: Springer Nature Singapore.
42. Ashok, J., RAMINENI, K., & Rajan, E. G. (2010). BEYOND INFORMATION RETRIEVAL: A SURVEY. *Journal of Theoretical & Applied Information Technology*, *15*.
43. Sekhar, P. R., & Sujatha, B. (2020, July). A literature review on feature selection using evolutionary algorithms. In *2020 7th International Conference on Smart Structures and Systems (ICSSS)* (pp. 1-8). IEEE.
44. Sekhar, P. R., & Sujatha, B. (2023). Feature extraction and independent subset generation using genetic algorithm for improved classification. *Int. J. Intell. Syst. Appl. Eng*, *11*, 503-512.
45. Sekhar, P. R., & Goud, S. (2024). Collaborative Learning Techniques in Python Programming: A Case Study with CSE Students at Anurag University. *Journal of Engineering Education Transformations*, *38*(Special Issue 1).
46. Pesaramelli, R. S., & Sujatha, B. (2024, March). Principle correlated feature extraction using differential evolution for improved classification. In *AIP Conference Proceedings* (Vol. 2919, No. 1). AIP Publishing.
47. Amarnadh, V., & Moparthi, N. R. (2023). Comprehensive review of different artificial intelligence-based methods for credit risk assessment in data science. *Intelligent Decision Technologies*, *17*(4), 1265-1282.
48. Amarnadh, V., & Moparthi, N. R. (2024). Prediction and assessment of credit risk using an adaptive Binarized spiking marine predators' neural network in financial sector. *Multimedia Tools and Applications*, *83*(16), 48761-48797.
49. Amarnadh, V., & Moparthi, N. R. (2024). Range control-based class imbalance and optimized granular elastic net regression feature selection for credit risk assessment. *Knowledge and Information Systems*, 1-30.
50. Amarnadh, V., & Akhila, M. (2019, May). RETRACTED: Big Data Analytics in E-Commerce User Interest Patterns. In *Journal of Physics: Conference Series* (Vol. 1228, No. 1, p. 012052). IOP Publishing.
51. Selvan, M. Arul, and S. Miruna Joe Amali. "RAINFALL DETECTION USING DEEP LEARNING TECHNIQUE." (2024).
52. Selvan, M. Arul. "Fire Management System For Indutrial Safety Applications." (2023).
53. Selvan, M. A. (2023). A PBL REPORT FOR CONTAINMENT ZONE ALERTING APPLICATION.
54. Selvan, M. A. (2023). CONTAINMENT ZONE ALERTING APPLICATION A PROJECT BASED LEARNING REPORT.
55. Selvan, M. A. (2021). Robust Cyber Attack Detection with Support Vector Machines: Tackling Both Established and Novel Threats.
56. Selvan, M. A. (2023). INDUSTRY-SPECIFIC INTELLIGENT FIRE MANAGEMENT SYSTEM.

57. Selvan, M. Arul. "PHISHING CONTENT CLASSIFICATION USING DYNAMIC WEIGHTING AND GENETIC RANKING OPTIMIZATION ALGORITHM." (2024).
58. Selvan, M. Arul. "Innovative Approaches in Cardiovascular Disease Prediction Through Machine Learning Optimization." (2024).
59. Lokhande, M., Kalpanadevi, D., Kate, V., Tripathi, A. K., & Bethapudi, P. (2023). Study of Computer Vision Applications in Healthcare Industry 4.0. In *Healthcare Industry 4.0* (pp. 151-166). CRC Press.
60. Tripathi, A. K., Soni, R., & Verma, S. (2022). A review on ethnopharmacological applications, pharmacological activities, and bioactive compounds of Mimosa pudica (linn.). *Research Journal of Pharmacy and Technology*, *15*(9), 4293-4299.
61. Mishra, S., Grewal, J., Wal, P., Bhivshet, G. U., Tripathi, A. K., & Walia, V. (2024). Therapeutic potential of vasopressin in the treatment of neurological disorders. *Peptides*, *174*, 171166.
62. Koliqi, R., Fathima, A., Tripathi, A. K., Sohi, N., Jesudasan, R. E., & Mahapatra, C. (2023). Innovative and Effective Machine Learning-Based Method to Analyze Alcoholic Brain Activity with Nonlinear Dynamics and Electroencephalography Data. *SN Computer Science*, *5*(1), 113.
63. Biswas, D., Sharma, G., Pandey, A., Tripathi, A. K., Pandey, A., & Sahu, P. & Chauhan, P.(2022). Magnetic Nanosphere: Promising approach to deliver the drug to the site of action. *NeuroQuantology*, *20*(11), 4038.
64. Tripathi, A. K., Diwedi, P., Kumar, N., Yadav, B. K., & Rathod, D. (2022). Trigonella Foenum Grecum L. Seed (Fenugreek) Pharmacological Effects on Cardiovascular and Stress Associated Disease. *NeuroQuantology*, *20*(8), 4599.
65. Tripathi, A. K., Dwivedi, C. P., Bansal, P., Pradhan, D. K., Parganiha, R., & Sahu, D. An Ethnoveterinary Important Plant Terminalia Arjuna. *International Journal of Health Sciences*, (II), 10601-10607.
66. Babbar, R., Kaur, A., Vanya, Arora, R., Gupta, J. K., Wal, P., ... & Behl, T. (2024). Impact of Bioactive Compounds in the Management of Various Inflammatory Diseases. Current Pharmaceutical Design, 30(24), 1880-1893.
67. Sahu, A., Mishra, S., Wal, P., Debnath, B., Chouhan, D., Gunjal, S. D., & Tripathi, A. K. (2024). Novel Quinoline-Based RAF Inhibitors: A Comprehensive Review on Synthesis, SAR and Molecular Docking Studies. *ChemistrySelect*, *9*(23), e202400347.
68. Vaishnav, Y., Banjare, L., Verma, S., Sharma, G., Biswas, D., Tripathi, A., ... & Manjunath, K. (2022). Computational Method on Hydroxychloroquine and Azithromycin for SARS-CoV-2: Binding Affinity Studies. *Research Journal of Pharmacy and Technology*, *15*(12), 5467-5472.
69. Ramya, S., Devi, R. S., Pandian, P. S., Suguna, G., Suganya, R., & Manimozhi, N. (2023). Analyzing Big Data challenges and security issues in data privacy. *International Research Journal of Modernization in Engineering Technology and Science*, *5*(2023), 421-428.
70. Pandian, P. S., & Srinivasan, S. (2016). A Unified Model for Preprocessing and Clustering Technique for Web Usage Mining. *Journal of Multiple-Valued Logic & Soft Computing*, *26*.
71. Thamma, S. R. T. S. R. (2025). Transforming E-Commerce with Pragmatic Advertising Using Machine Learning Techniques.
72. Thamma, S. R. T. S. R. (2024). Optimization of Generative AI Costs in Multi-Agent and Multi-Cloud Systems.
73. Thamma, S. R. T. S. R. (2024). Revolutionizing Healthcare: Spatial Computing Meets Generative AI.
74. Thamma, S. R. T. S. R. (2024). Cardiovascular image analysis: AI can analyze heart images to assess cardiovascular health and identify potential risks.
75. Thamma, S. R. T. S. R. (2024). Generative AI in Graph-Based Spatial Computing: Techniques and Use Cases.
76. NAVANEETHA, N., & KALYANI, S. (2012). Efficient Association Rule Mining using Indexing Support.
77. Thirumoorthi, P., Deepika, S., & Yadaiah, N. (2014, March). Solar energy based dynamic sag compensator. In *2014 International Conference on Green Computing Communication and Electrical Engineering (ICGCCEE)* (pp. 1-6). IEEE.
78. Nair, R., Zafrullah, S. N., Vinayasree, P., Singh, P., Zahra, M. M. A., Sharma, T., & Ahmadi, F. (2022). Blockchain-Based Decentralized Cloud Solutions for Data Transfer. *Computational Intelligence and Neuroscience*, *2022*(1), 8209854.
79. Vinayasree, P., & Reddy, A. M. (2023). Blockchain-Enabled Hyperledger Fabric to Secure Data Transfer Mechanism for Medical Cyber-Physical System: Overview, Issues, and Challenges. *EAI Endorsed Transactions on Pervasive Health and Technology*, *9*.
80. Vinayasree, P., & Reddy, A. M. (2025). A Reliable and Secure Permissioned Blockchain-Assisted Data Transfer Mechanism in Healthcare-Based Cyber-Physical Systems. *Concurrency and Computation: Practice and Experience*, *37*(3), e8378.

81. VINAYASREE[1], P., & REDDY, A. M. (2024). A SCALABLE AND SECURE BLOCKCHAIN-BASED HEALTHCARE SYSTEM: OPTIMIZING PERFORMANCE, SECURITY, AND PRIVACY WITH ADAPTIVE TECHNOLOGIES. *Journal of Theoretical and Applied Information Technology*, *102*(22).
82. Sahoo, P. K., & Jeripothula, P. (2020). Heart failure prediction using machine learning techniques. *Available at SSRN 3759562*.
83. Sahoo, P. K., Chottray, R. K., & Pattnaiak, S. (2012). Research issues on windows event log. *International Journal of Computer Applications*, *41*(19).
84. Sahoo, P. K. (2018, March). Data mining a way to solve Phishing Attacks. In *2018 International Conference on Current Trends towards Converging Technologies (ICCTCT)* (pp. 1-5). IEEE.
85. Sahoo, P. K., Chhotray, R. K., Jena, G., & Pattnaik, S. (2013). An implementation of elliptic curve cryptography. *Int. J. Eng. Res. Technol.(IJERT)*, *2*(1), 2278-0181.
86. Nagesh, O., Kumar, T., & Venkateswararao, V. (2017). A Survey on Security Aspects of Server Virtualization in Cloud Computing. *International Journal of Electrical & Computer Engineering (2088-8708)*, *7*(3).
87. Budaraju, R. R., & Nagesh, O. S. (2023, June). Multi-Level Image Thresholding Using Improvised Cuckoo Search Optimization Algorithm. In *2023 3rd International Conference on Intelligent Technologies (CONIT)* (pp. 1-7). IEEE.
88. Nagesh, O. S., Budaraju, R. R., Kulkarni, S. S., Vinay, M., Ajibade, S. S. M., Chopra, M., ... & Kaliyaperumal, K. (2024). Boosting enabled efficient machine learning technique for accurate prediction of crop yield towards precision agriculture. *Discover Sustainability*, *5*(1), 78.
89. Jyothi, A., & Indira, B. (2018). A Two Way Validation Framework for Cloud Storage Security. *International Journal of Engineering & Technology*, *7*(2.20), 236-242.
90. Rekha, S. B., & Rao, M. V. (2017, September). Methodical activity recognition and monitoring of a person through smart phone and wireless sensors. In *2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI)* (pp. 1456-1459). IEEE.
91. Sangisetti, B. R., Pabboju, S., & Racha, S. (2019, June). Smart call forwarding and conditional signal monitoring in duos mobile. In *Proceedings of the Third International Conference on Advanced Informatics for Computing Research* (pp. 1-11).
92. Sangisetti, B. R., & Pabboju, S. (2021). Analysis on human activity recognition using machine learning algorithm and personal activity correlation. *Psychol Educ J*, *58*(2), 5754-5760.
93. Kumar, T. V. (2018). Project Risk Management System Development Based on Industry 4.0 Technology and its Practical Implications.
94. Tambi, V. K., & Singh, N. (2015). Potential Evaluation of REST Web Service Descriptions for Graph-Based Service Discovery with a Hypermedia Focus.
95. Kumar, T. V. (2024). A Comparison of SQL and NO-SQL Database Management Systems for Unstructured Data.
96. Kumar, T. V. (2024). A Comprehensive Empirical Study Determining Practitioners' Views on Docker Development Difficulties: Stack Overflow Analysis.
97. Kumar, T. V. (2024). Developments and Uses of Generative Artificial Intelligence and Present Experimental Data on the Impact on Productivity Applying Artificial Intelligence that is Generative.
98. Kumar, T. V. (2024). A New Framework and Performance Assessment Method for Distributed Deep Neural NetworkBased Middleware for Cyberattack Detection in the Smart IoT Ecosystem.
99. Sharma, S., & Dutta, N. (2016). Analysing Anomaly Process Detection using Classification Methods and Negative Selection Algorithms.
100. Sharma, S., & Dutta, N. (2024). Examining ChatGPT's and Other Models' Potential to Improve the Security Environment using Generative AI for Cybersecurity.
101. Sakshi, S. (2023). Development of a Project Risk Management System based on Industry 4.0 Technology and its Practical Implications.
102. Arora, P., & Bhardwaj, S. Mitigating the Security Issues and Challenges in the Internet of Things (IOT) Framework for Enhanced Security.
103. Sakshi, S. (2024). A Large-Scale Empirical Study Identifying Practitioners' Perspectives on Challenges in Docker Development: Analysis using Stack Overflow.
104. Sakshi, S. (2023). Advancements and Applications of Generative Artificial Intelligence and show the Experimental Evidence on the Productivity Effects using Generative Artificial Intelligence.
105. Sakshi, S. (2023). Assessment of Web Services based on SOAP and REST Principles using Different Metrics for Mobile Environment and Multimedia Conference.
106. Sakshi, S. (2022). Design and Implementation of a Pattern-based J2EE Application Development Environment.

107. Sharma, S., & Dutta, N. (2018). Development of New Smart City Applications using Blockchain Technology and Cybersecurity Utilisation. Development, 7(11).
108. Sharma, S., & Dutta, N. (2017). Development of Attractive Protection through Cyberattack Moderation and Traffic Impact Analysis for Connected Automated Vehicles. Development, 4(2).
109. Sharma, S., & Dutta, N. (2015). Evaluation of REST Web Service Descriptions for Graph-based Service Discovery with a Hypermedia Focus. Evaluation, 2(5).
110. Sharma, S., & Dutta, N. (2024). Examining ChatGPT's and Other Models' Potential to Improve the Security Environment using Generative AI for Cybersecurity.
111. Sharma, S., & Dutta, N. (2015). Cybersecurity Vulnerability Management using Novel Artificial Intelligence and Machine Learning Techniques. Sakshi, S. (2023). Development of a Project Risk Management System based on Industry 4.0 Technology and its Practical Implications.
112. Sharma, S., & Dutta, N. (2017). Classification and Feature Extraction in Artificial Intelligence-based Threat Detection using Analysing Methods.
113. Sharma, S., & Dutta, N. (2016). Analysing Anomaly Process Detection using Classification Methods and Negative Selection Algorithms.
114. Sharma, S., & Dutta, N. (2015). Distributed DNN-based Middleware for Cyberattack Detection in the Smart IOT Ecosystem: A Novel Framework and Performance Evaluation Technique.
115. Bhat, S. (2015). Technology for Chemical Industry Mixing and Processing. Technology, 2(2).
116. Bhat, S. (2024). Building Thermal Comforts with Various HVAC Systems and Optimum Conditions.
117. Bhat, S. (2020). Enhancing Data Centre Energy Efficiency with Modelling and Optimisation of End-To-End Cooling.
118. Bhat, S. (2016). Improving Data Centre Energy Efficiency with End-To-End Cooling Modelling and Optimisation.
119. Bhat, S. (2015). Deep Reinforcement Learning for Energy-Saving Thermal Comfort Management in Intelligent Structures.
120. Bhat, S. (2015). Design and Function of a Gas Turbine Range Extender for Hybrid Vehicles.
121. Bhat, S. (2023). Discovering the Attractiveness of Hydrogen-Fuelled Gas Turbines in Future Energy Systems.
122. Bhat, S. (2019). Data Centre Cooling Technology's Effect on Turbo-Mode Efficiency.
123. Bhat, S. (2018). The Impact of Data Centre Cooling Technology on Turbo-Mode Efficiency.
124. Archana, B., & Sreedaran, S. (2023). Synthesis, characterization, DNA binding and cleavage studies, in-vitro antimicrobial, cytotoxicity assay of new manganese (III) complexes of N-functionalized macrocyclic cyclam based Schiff base ligands. Polyhedron, 231, 116269.
125. Archana, B., & Sreedaran, S. (2022). New cyclam based Zn (II) complexes: effect of flexibility and para substitution on DNA binding, in vitro cytotoxic studies and antimicrobial activities. Journal of Chemical Sciences, 134(4), 102.
126. Archana, B., & Sreedaran, S. (2021). POTENTIALLY ACTIVE TRANSITION METAL COMPLEXES SYNTHESIZED AS SELECTIVE DNA BINDING AND ANTIMICROBIAL AGENTS. European Journal of Molecular and Clinical Medicine, 8(1), 1962-1971.
127. Rasappan, A. S., Palanisamy, R., Thangamuthu, V., Dharmalingam, V. P., Natarajan, M., Archana, B., ... & Kim, J. (2024). Battery-type WS2 decorated WO3 nanorods for high-performance supercapacitors. Materials Letters, 357, 135640.
128. Arora, P., & Bhardwaj, S. (2017). Investigation and Evaluation of Strategic Approaches Critically before Approving Cloud Computing Service Frameworks.
129. Arora, P., & Bhardwaj, S. (2017). Enhancing Security using Knowledge Discovery and Data Mining Methods in Cloud Computing.
130. Arora, P., & Bhardwaj, S. (2017). Combining Internet of Things and Wireless Sensor Networks: A Security-based and Hierarchical Approach.
131. Arora, P., & Bhardwaj, S. (2019). Safe and Dependable Intrusion Detection Method Designs Created with Artificial Intelligence Techniques. machine learning, 8(7).
132. Arora, P., & Bhardwaj, S. (2017). A Very Safe and Effective Way to Protect Privacy in Cloud Data Storage Configurations.
133. Arora, P., & Bhardwaj, S. (2019). The Suitability of Different Cybersecurity Services to Stop Smart Home Attacks.
134. Arora, P., & Bhardwaj, S. (2020). Research on Cybersecurity Issues and Solutions for Intelligent Transportation Systems.
135. Arora, P., & Bhardwaj, S. (2021). Methods for Threat and Risk Assessment and Mitigation to Improve Security in the Automotive Sector. Methods, 8(2).