MAJOR PROJECT REPORT (MSIC434)

Submitted to the University of Madras in partial fulfilment of the requirements for the award of the degree.

WELL-AI Personalised AI-based health Recommender.

Submitted by

VENKATESH P - 36823025 (MCA II)

Under the guidance of

Professor: Dr. Dr S. GOPINATHAN,

Department of Computer Science

UNIVERSITY OF MADRAS

GUINDY, CHENNAI - 600 025

DEPARTMENT OF COMPUTER SCIENCE

UNIVERSITY OF MADRAS
GUINDY, CHENNAI - 600 025

DEPARTMENT OF COMPUTER SCIENCE

BONAFIDE CERTIFICATE

This document serves to officially certify the substantial project work titled "**WELL-AI: Personalised AI-based Health Recommender**" submitted to the University of Madras, Guindy, by **VENKATESH P** (Regno: 36823025). This work was partially submitted in partial fulfilment of the requirements for the award of the **Master of Computer Applications** (MCA) degree. The document serves as a comprehensive record of the project work undertaken by the individual during the academic period **from December 2024 to April** 2025.

PROJECT GUIDE
Dr. S. Gopinathan

HEAD OF THE DEPARTMENT
Dr. S. Gopinathan

Submitted for the project Examination held on _____ at the Department of Computer Science, University of Madras, Guindy, Chennai – 600 025.

Internal Examiner

External Examiner

ACKNOWLEDGEMENT

We express our profound gratitude for our contributions to the project's successful completion.

We extend our sincere appreciation to **PROF. Dr S. GOPINATHAN (PROJECT GUIDE),** for her invaluable guidance, encouragement, and expert insights throughout the project.

We respect and thank **PROF. Dr S. GOPINATHAN (HEAD OF THE DEPARTMENT)** for the encouragement extended.

Without their unwavering support and mentorship, the project would not have attained its current state. We are deeply appreciative of their unwavering trust in our capabilities.

DECLARATION

I hereby declare that the Major Project entitled "**WELL-AI: Personalised AI-based Health Recommender"** is submitted by me for the award of the degree of MCA in Computer Science from the University of Madras is a record of project work done by me during the period DECEMBER 2024 - APRIL 2025 and the Major Project has not formed the basis for the award of any degree or other similar titles.

Student.
(P Venkatesh)

CONTEXT

1. ABSTRACT

In the rapidly evolving digital landscape, the healthcare and wellness sector continues to be a critical domain that urgently requires technological advancements. **Well-AI**, a personalized health recommendation web application, addresses this need by seamlessly integrating **artificial intelligence**, contemporary **web technologies**, and user-friendly **UI/UX design**. Developed using **ReactJS**, **TailwindCSS**, **Firebase**, **FastAPI**, and **scikit-learn**, Well-AI empowers users to effectively monitor and manage their health. The backend employs trained machine learning models (via joblib) to provide precise and personalized health recommendations, while Firebase Authentication ensures secure user access. Equipped with a clean and mobile-first responsive UI, Well-AI strives to bridge the gap between users and accessible and intelligent healthcare tools.

2. INTRODUCTION

With the rise in awareness regarding physical and mental health, the need for a centralized platform offering personalized and AI-driven health insights has become more evident. **Well-AI** serves this purpose by offering modules for general health, mental wellness, and AI-powered symptom analysis. Users can access secure features through Firebase Authentication and experience an engaging, intuitive interface built with modern front-end technologies like React and Tailwind CSS. This project demonstrates how health tech can be enhanced using scalable web development practices and machine learning models.

3. ORGANIZATIONAL PROFILE

NASO Technology is a dynamic tech organization dedicated to driving innovation in AI, cloud computing, and web development. Renowned for its support of student projects and initiatives in advanced technologies, NASO offers essential mentorship and robust infrastructure. This environment empowers young engineers to develop scalable, impactful solutions. The project highlighted here exemplifies NASO's commitment to harnessing technology for significant societal benefits, showcasing their role in nurturing future tech leaders.

4. SYSTEM SPECIFICATION

Category	Requirements
Hardware	Processor: Intel i5/i7 or equivalent RAM: 8GB minimum Storage: 2GB available disk space GPU: Optional for model training (not required in deployment)
Software	OS: Windows 10/11, Ubuntu 20.04+, or macOS Code Editor: VS Code Backend: Python 3.10+, FastAPI Frontend: Node.js 18+, React.js, Tailwind CSS Database: Firebase Firestore Libraries: scikit-learn, joblib, Firebase SDK

5. PROBLEM STATEMENT

In today's fast-paced world, individuals face increasing challenges in monitoring both their physical and mental well-being. This struggle largely stems from the absence of centralized, intelligent, and user-friendly platforms that can comprehensively address their health tracking needs. Traditional health check systems often rely on manual processes, making them time-consuming and less efficient. Additionally, many of these systems are either inaccessible to a broad audience due to technological barriers or overly generalized, failing to cater to individual health requirements.

Recognizing these limitations, this project is designed to bridge the gap by introducing an intelligent, responsive, and personalized health tracking platform. Leveraging the power of artificial intelligence (AI) and modern web technologies, the platform aims to provide users with tailored insights and actionable health data. This innovative approach ensures that users can easily monitor and manage their health, promoting a more holistic and proactive approach to well-being.

6. METHODOLOGIES

Agile Development: Agile development emphasizes iterative progress through short, manageable sprints, allowing teams to adapt quickly to changes. Continuous integration ensures that code changes are frequently merged, tested, and deployed, reducing integration issues and enhancing software quality.

Component-Based Architecture: A modular frontend architecture using React promotes reusability, scalability, and maintainability. By breaking down the UI into independent, self-contained components, developers can efficiently manage complex applications, streamline updates, and improve code organization.

RESTful API Design: Backend development with FastAPI facilitates building high-performance APIs adhering to REST principles. FastAPI's simplicity and speed enable efficient handling of HTTP requests, seamless data exchange, and robust endpoint management, supporting scalable applications.

Firebase Integration: Leveraging Firebase for authentication and database management offers secure, real-time data synchronization and user management. Its easy-to-implement features allow for rapid development, seamless cross-platform support, and reliable backend services, enhancing application performance.

Machine Learning Pipeline: Implementing machine learning pipelines with scikit-learn involves training models and deploying them using joblib. This method ensures efficient model serialization, quick loading during production, and streamlined workflows for predictive analytics and data-driven decision-making.

7. SOFTWARE DESCRIPTION

Well-AI is a comprehensive application designed to deliver advanced functionality through a combination of modern technologies across its frontend, backend, database, authentication, and machine learning components.

Frontend: The user interface of Well-AI is crafted with React, a powerful JavaScript library known for building interactive and dynamic UIs. To ensure the design is sleek and responsive, Tailwind CSS is employed, offering utility-first styling that allows for rapid UI development. Additionally, Framer Motion is integrated to add smooth, visually appealing animations, enhancing user experience through dynamic transitions and effects.

Backend: The backend infrastructure is robustly developed using FastAPI, a high-performance web framework for building APIs with Python. FastAPI effectively manages user requests, ensuring quick responses and efficient data handling. It also serves as the integration layer for machine learning (ML) models, facilitating seamless communication between the frontend and the predictive algorithms.

Database: Well-AI utilizes Firebase Firestore, a flexible and scalable NoSQL cloud database, to store and manage user data along with prediction records. Firestore's real-time synchronization feature ensures that data is updated instantaneously across all platforms.

Authentication: For secure user management, Well-AI relies on Firebase Authentication. This service handles user sign-ups, logins, and session management, ensuring secure and seamless access to the application.

Machine Learning Models: At its core, Well-AI incorporates machine learning models built using Scikit-learn, a popular Python library. These models are designed for symptom-based predictions, providing users with insights based on their input data, thus enhancing the app’s diagnostic capabilities.

8. WORKFLOW

User Journey Through the Health Application

Landing on the Welcome Page: The user's experience begins when they visit the application's welcome page. This serves as the main entry point, designed to offer a clear overview of the app's features and intuitive navigation options. The welcome page aims to engage the user with its user-friendly design, guiding them toward the next steps seamlessly.

Secure Sign-In via Firebase Authentication: To maintain secure access, the application prompts the user to sign in using Firebase Authentication. This process verifies the user's credentials, ensuring their data's safety and maintaining session security. Firebase Authentication supports multiple sign-in methods, including email/password, social logins, and more, enhancing accessibility.

Accessing the Comprehensive Dashboard: After successful authentication, the user is directed to the dashboard. This central hub presents a variety of health-related modules:

General Health: Offers tools and resources for tracking and improving physical well-being, including activity logs, dietary suggestions, and health metrics.

Mental Health: Provides mental health assessments, mindfulness tips, stress management techniques, and access to professional resources.

Symptom-Based Prediction: Enables users to input specific symptoms, facilitating early detection and predictive analysis of possible health conditions.

Data Submission to FastAPI Backend: When a user selects a module and inputs relevant data, this information is securely

transmitted to the FastAPI backend. FastAPI efficiently handles the data, ensuring quick processing and minimal latency.

Machine Learning Model Analysis: The FastAPI backend communicates with an integrated machine learning model. This model processes the user input, leveraging trained algorithms to generate accurate health-related predictions.

Receiving Results and Tailored Recommendations: Finally, the user receives the analyzed results accompanied by personalized recommendations. These insights help users understand their health status better and suggest actionable next steps, empowering them to make informed decisions about their well-being.

9. COMPONENTS OF THE PROJECT

The application structure consists of several key frontend pages, common elements, and backend components designed to provide a seamless user experience while delivering accurate health-related predictions.

Frontend Pages:

Welcome Page: Serves as the introductory interface, offering an engaging overview of the application's purpose and features to new and returning users.
Login/Signup Page: Facilitates secure user authentication, providing forms for new user registration and returning user login, ensuring data privacy and protection.
Loading Page: Acts as a transitional screen, indicating the processing of data or navigation between pages to enhance the user flow.
General Health Page: Offers insights into overall health metrics, displaying personalized recommendations and health tips based on user data.
Mental Health Page: Focuses on mental well-being, providing resources, assessments, and support tools tailored to users' mental health requirements.
Symptom-Based Prediction Page: Enables users to input symptoms to receive predictive analysis, leveraging machine learning algorithms for health condition forecasts.

Common Elements:

Navigation Bar: Provides easy access to different sections of the application, ensuring intuitive movement across pages.
Auth Components: Includes reusable authentication modules to manage user sessions securely.

Responsive Design Modules: Ensures that the application is accessible and user-friendly across various device types and screen sizes.

Backend Components:

API Endpoints: Designed to handle health data submissions, manage user requests, and deliver predictive analytics results efficiently.
ML Model Loading with Joblib: Facilitates the integration of trained machine learning models into the application, allowing for swift and accurate predictions.
Firebase Integration Logic: Manages real-time database operations, user authentication, and cloud storage, enhancing the application's performance and scalability.

This architecture ensures a robust, user-centric application capable of addressing diverse health-related concerns effectively.

10 SAMPLE CODE

FRONT END SAMPLE

```
import { useEffect, useState } from "react";
import { BrowserRouter as Router, Routes, Route } from "react-router-dom";
import { onAuthStateChanged } from "firebase/auth";
import { auth } from "./firebase";
import Welcome from "./pages/Welcome";
import Login from "./auth/Login";
import Signup from "./auth/Signup";
import Loading from "./pages/Loading";
import Dashboard from './pages/Dashboard';
import MentalHealth from "./pages/MentalHealth";
import SymptomBased from "./pages/SymptomsBased";
import Settings from "./pages/Settings";
import FitbitCallback from "./pages/FitbitCallback";
import { ThemeProvider } from "./pages/ThemeContext";
const App = () => {
  const [loading, setLoading] = useState(true);
  const [user, setUser] = useState(null);

  useEffect(() => {
    const unsubscribe = onAuthStateChanged(auth, (currentUser) => {
      setUser(currentUser);
      setLoading(false);
    });

    return () => unsubscribe();
  }, []);

  if (loading) return <Loading />;
  return (
    <ThemeProvider>
    <Router>
      <Routes>
      <Route path="/" element={<Welcome />} />
        <Route path="/login" element={<Login />} />
        <Route path="/signup" element={<Signup />} />
          <Route path="Dashboard" element={<Dashboard />} />
            <Route path="MentalHealth" element={<MentalHealth />} />
            <Route path="SymptomBased" element={<SymptomBased />} />
            <Route path="settings" element={<Settings />} />
            <Route path="/fitbit/callback" element={<FitbitCallback />} />
        {/* Add more routes here */}
      </Routes>
    </Router>
    </ThemeProvider>
```

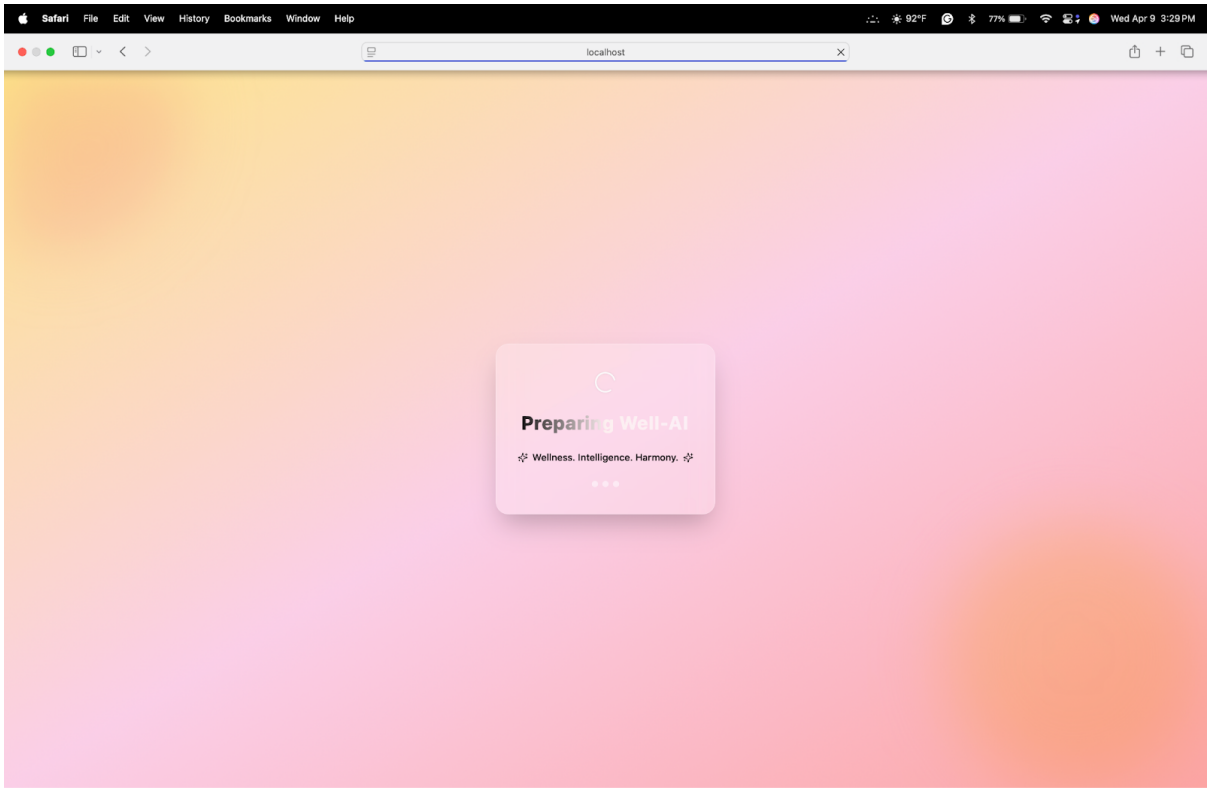
```
  );
};
export default App;                                                    APP.JS
```

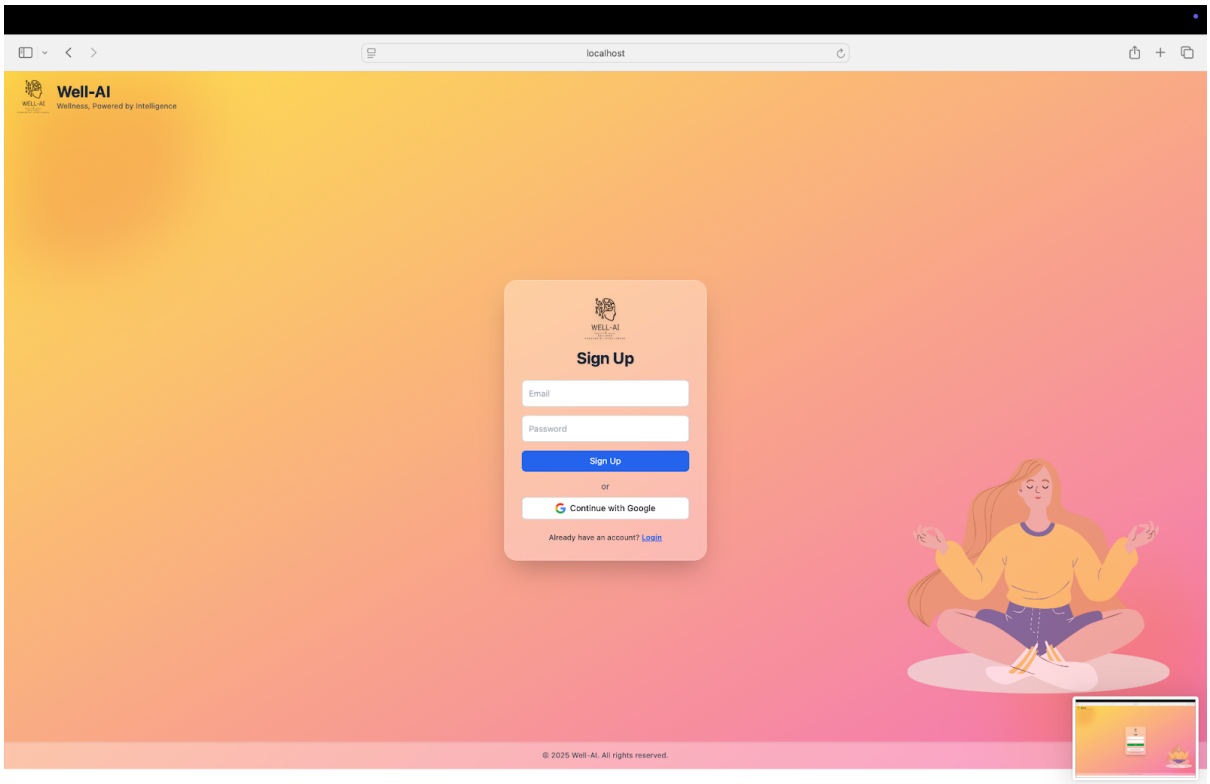
BACKEND SAMPLE

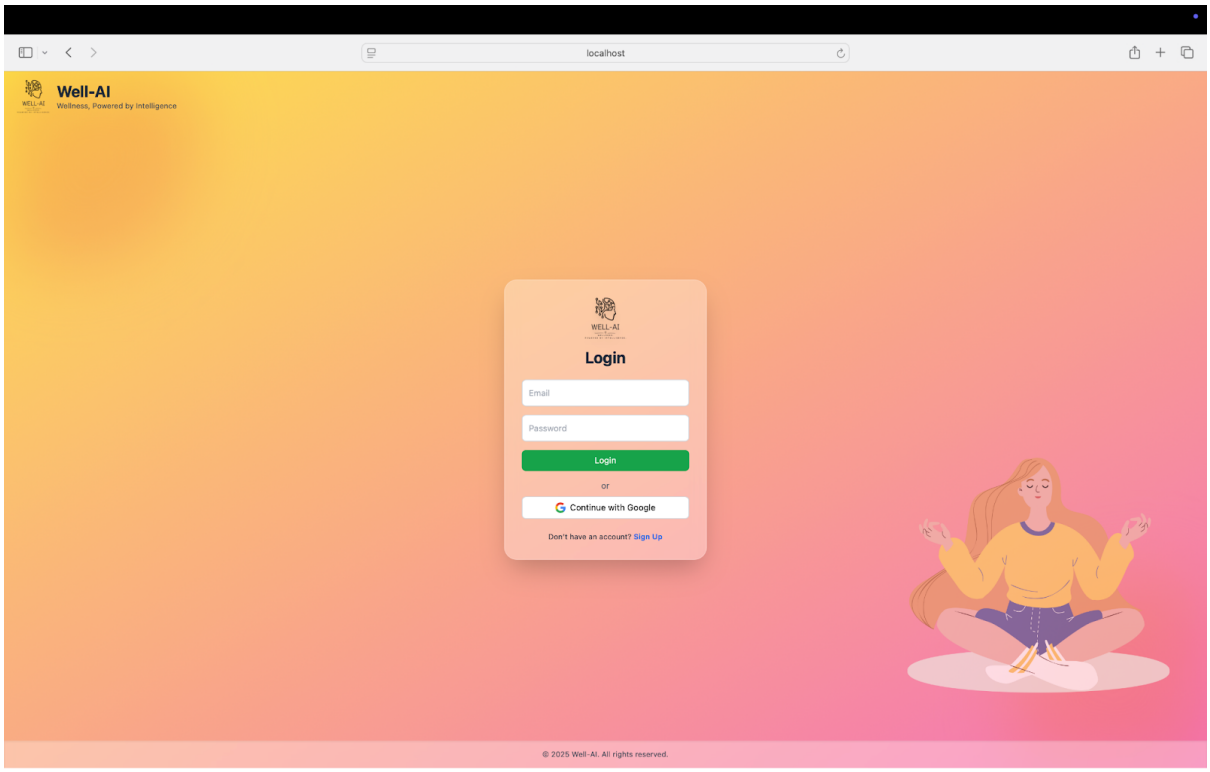
```
from fastapi import FastAPI, HTTPException
from fastapi.middleware.cors import CORSMiddleware
from pydantic import BaseModel
import joblib
import numpy as np
import pandas as pd

app = FastAPI()

# Enable CORS
app.add_middleware(CORSMiddleware, allow_origins=["*"], allow_credentials=True,
allow_methods=["*"], allow_headers=["*"])

# Request Models
class SymptomRequest(BaseModel):
    symptoms: list[str]

class DepressionRequest(BaseModel):
    Gender: str
    Age: int
    Sleep_Duration: str
    Work_Study_Hours: int
    Financial_Stress: int
    Academic_Work_Pressure: int
    Job_Study_Satisfaction: int
    Family_History_of_Mental_Illness: str
    Suicidal_Thoughts: str

# Load Models
svm_model = joblib.load("models/svm_8020.pkl")
depression_model = joblib.load("models/depression_model.pkl")
scaler = joblib.load("models/scaler.pkl")

# Symptom List
symptoms_list = ['itching', 'skin_rash', 'nodal_skin_eruptions', 'continuous_sneezing',
'chills', 'joint_pain']  # Shortened for simplicity

# Encode Symptoms
def encode_symptoms(symptoms):
    return np.array([1 if s in symptoms else 0 for s in symptoms_list]).reshape(1, -1)

# Preprocess Depression Data
def preprocess_depression(data: DepressionRequest):
    mapping = {"Male": 1, "Female": 0, "Yes": 1, "No": 0}
    sleep_hours = {"Less than 5 hours": 4, "5-6 hours": 5.5, "7-8 hours": 7.5, "More
than 8 hours": 9}
```

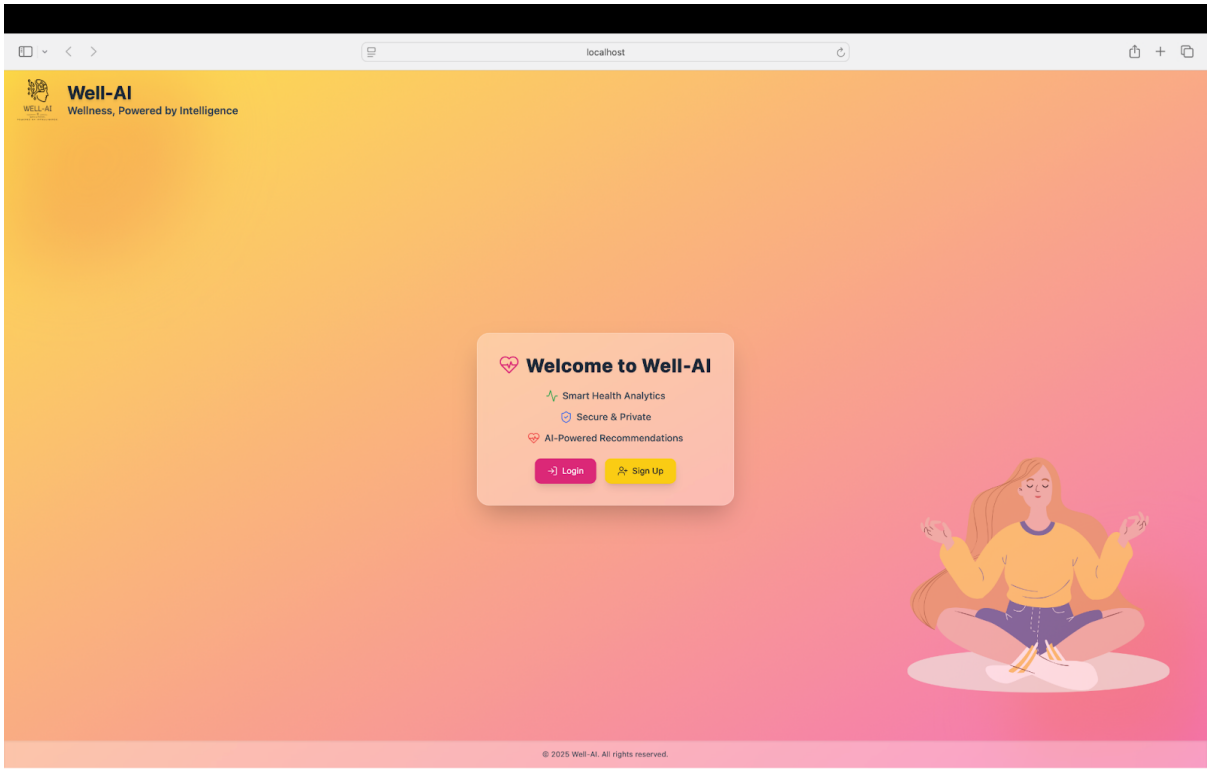
```
    df = pd.DataFrame([{
        "Gender": mapping.get(data.Gender, 0),
        "Age": data.Age,
        "Sleep Duration": sleep_hours.get(data.Sleep_Duration, 0),
        "Work/Study Hours": data.Work_Study_Hours,
        "Financial Stress": data.Financial_Stress,
        "Academic/Work Pressure": data.Academic_Work_Pressure,
        "Job/Study Satisfaction": data.Job_Study_Satisfaction,
        "Family History": mapping.get(data.Family_History_of_Mental_Illness, 0),
        "Suicidal Thoughts": mapping.get(data.Suicidal_Thoughts, 0)
    }])
    return scaler.transform(df)

# Routes
@app.post("/predict")
def predict_disease(req: SymptomRequest):
    if not req.symptoms:
        raise HTTPException(status_code=400, detail="Symptoms required")
    prediction = svm_model.predict(encode_symptoms(req.symptoms))[0]
    return {"prediction": prediction}

@app.post("/predict_depression")
def predict_depression(data: DepressionRequest):
    prediction = depression_model.predict(preprocess_depression(data))[0]
    return {"prediction": int(prediction)}
```

11. SAMPLE SCREENSHOTS

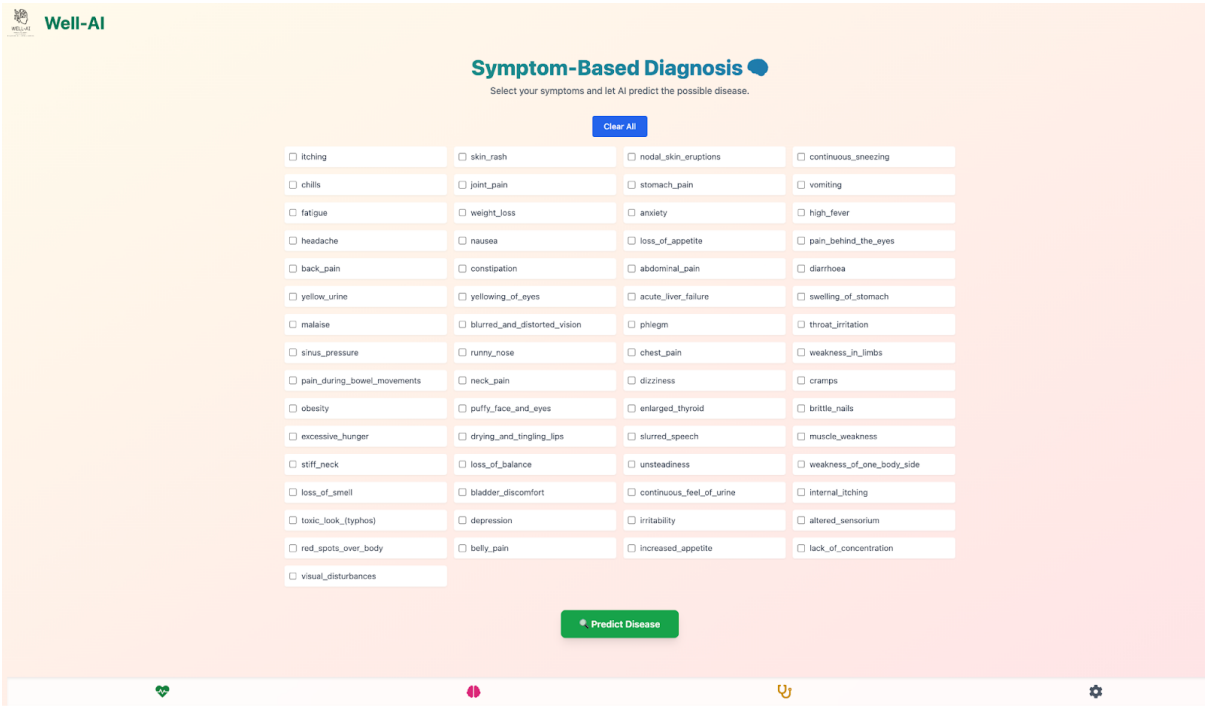


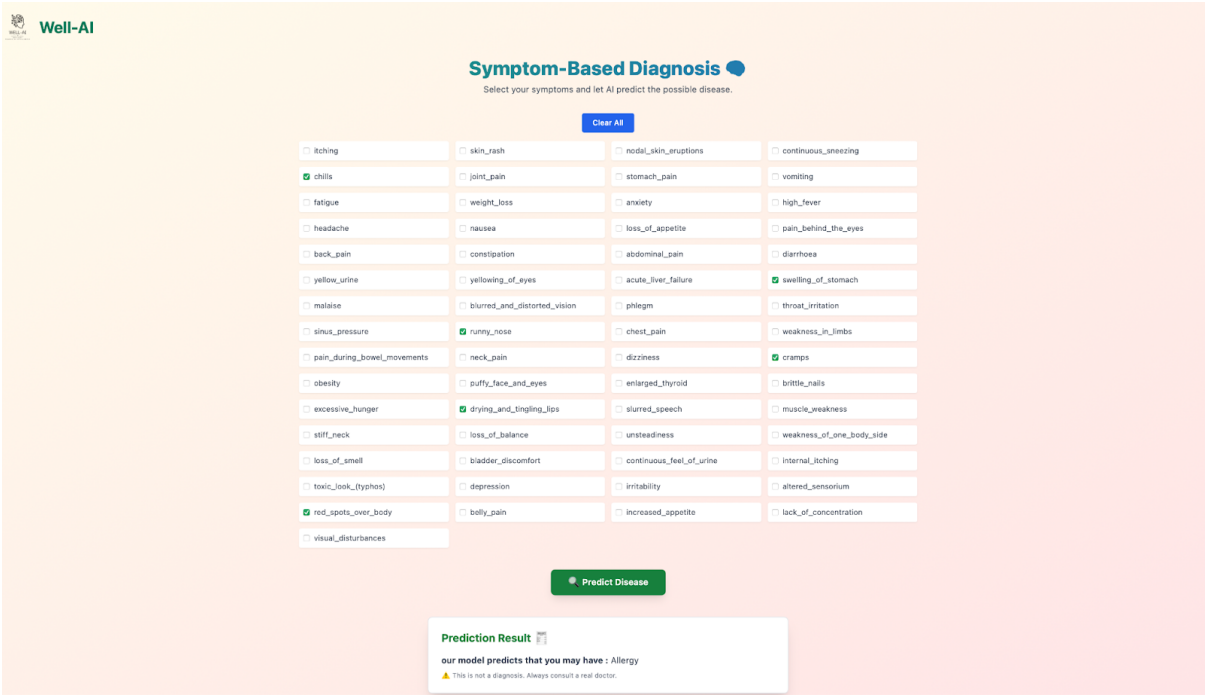



localhost
Well-AI
Wellness, Powered by Intelligence
WELL-AI
Login
Email
Password
Login
or
Continue with Google
Don't have an account? Sign Up
© 2025 Well-AI. All rights reserved.


localhost
Well-AI
Wellness, Powered by Intelligence
Welcome to Well-AI
Smart Health Analytics
Secure & Private
AI-Powered Recommendations
Login
Sign Up
© 2025 Well-AI. All rights reserved.

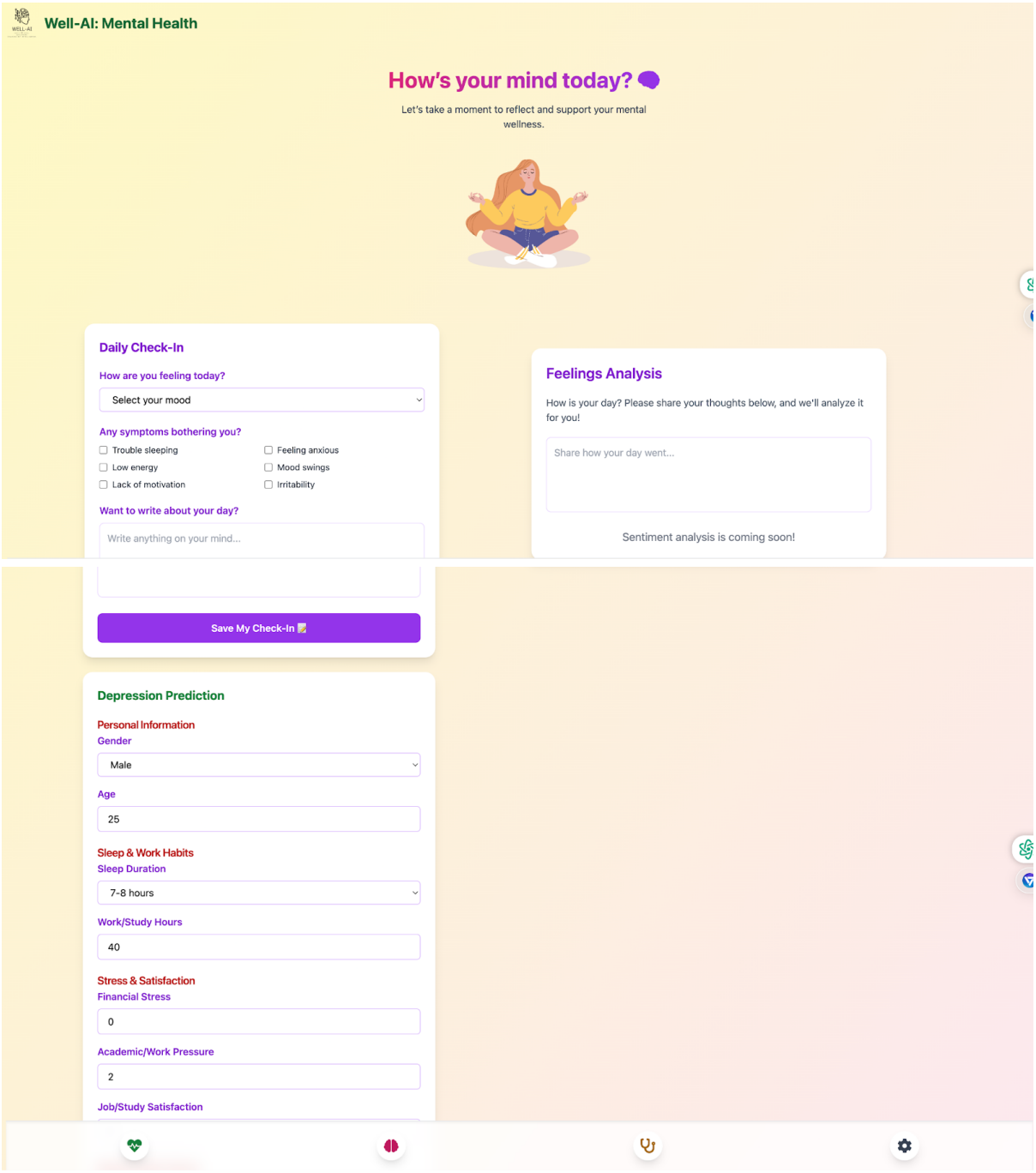
Symptoms prediction page



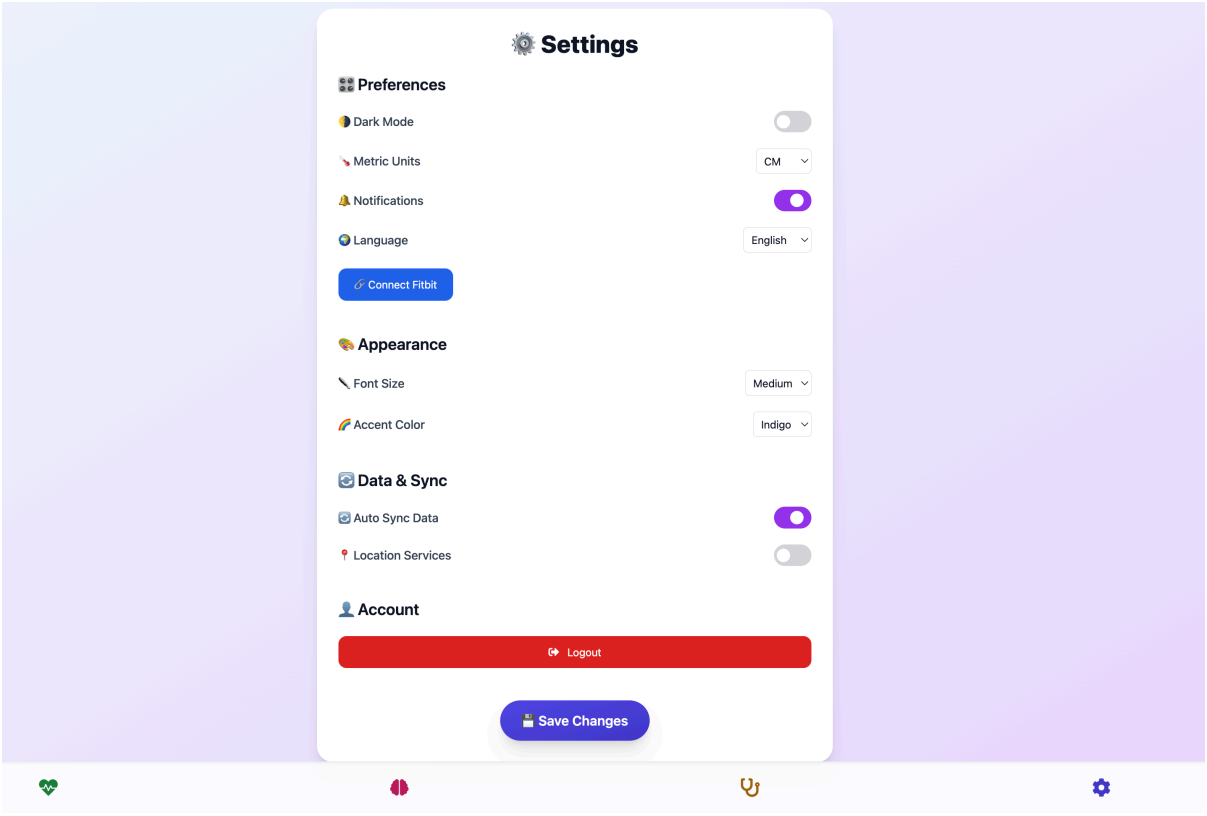


Its predict very well :-

Mental health page



Settings page:-



12. CONCLUSION

Well-AI is revolutionizing the healthcare landscape by effectively bridging the gap between cutting-edge technology and essential healthcare services. This innovative platform offers a scalable and intelligent solution designed for comprehensive health monitoring, catering to both individual users and healthcare providers. By harnessing the power of advanced artificial intelligence models, Well-AI delivers accurate, real-time insights into various health metrics, enabling proactive health management and early detection of potential health issues.

The platform is built with a strong emphasis on modern user interface (UI) principles, ensuring a seamless, intuitive, and user-friendly experience. Its design prioritizes ease of use, making it accessible to people of all ages and technological proficiencies. This practicality extends to its integration capabilities, allowing it to work harmoniously with existing healthcare systems and devices.

Well-AI's robust analytics not only track and analyze health data but also provide personalized recommendations, fostering informed decision-making for users. Whether it's monitoring vital signs, tracking fitness goals, or managing chronic conditions, Well-AI stands out as a versatile tool in everyday health management. By combining state-of-the-art technology with practical healthcare applications, Well-AI represents a significant step forward in promoting healthier lifestyles and improving overall well-being.

13. FUTURE ENHANCEMENT

- The integration of a GPT-based health chatbot transforms patient engagement and healthcare support.
- The chatbot provides personalized health advice, answers medical queries, and guides users through initial symptom assessments.
- Natural language processing ensures seamless, human-like interactions, enhancing user experience and fostering trust.
- A symptom tracking feature allows users to log health conditions with specificity, capturing variables like symptom onset, intensity, duration, and triggers.
- Granular data collection aids users in monitoring their health and provides healthcare professionals with valuable insights for accurate diagnostics and personalized treatment plans.
- A real-time analytics dashboard offers healthcare providers immediate access to patient data trends and health metrics.
- The dashboard supports data-driven decision-making, enabling quick identification of potential health issues, outbreak patterns, or patient-specific concerns.
- An intuitive design ensures medical staff can efficiently interpret the data, leading to improved patient outcomes.
- A mobile application developed using React Native provides accessibility and user convenience.
- The mobile app integrates all features—chatbot interaction, symptom tracking, and health analytics—into a unified platform.
- The app performs seamlessly on both iOS and Android devices, providing a consistent user experience.

14. REFERENCES

Here is an expanded list with referral links to the mentioned documentation and resources:

- **ReactJS Official Documentation** - Comprehensive guide to building user interfaces with React.https://react.dev/
- **Tailwind CSS Docs** - Official documentation for utility-first CSS framework, Tailwind CSS.https://tailwindcss.com/
- **Firebase Documentation** - Guides and API references for Google's Firebase platform.https://firebase.google.com/docs
- **FastAPI Docs** - Official documentation for FastAPI, a modern, fast web framework for building APIs with Python.https://fastapi.tiangolo.com/
- **Scikit-learn Library** - User guide for Scikit-learn, a Python module for machine learning.https://scikit-learn.org/stable/
- **Joblib Documentation** - Official documentation for Joblib, used for lightweight pipelining in Python https://pypi.org/project/joblib/
- **GitHub Repositories** - Explore repositories and codebases hosted on GitHub. https://github.com/venkatesh-hyper/Well.AI
- **StackOverflow Threads** - Explore discussions, Q&A threads, and community support on StackOverflow. https://stackoverflow.com/questions/62908873/how-to-add-firebase-ml-kit-in-react-native

- Feel free to click on the links to access the resources directly.

KAVIGAI
Explore | Discover | Succeed

Nas

Certificate of Completion

TO WHOMSOEVER IT MAY CONCERN

This is to certify that **Mr. VENKATESH P**, Reg. No.: 36823025, Mast Applications, **University of Madras, Guindy Campus** - 600 085 has complete Internship at **NASO Technologies Pvt Ltd., Chennai** from **January 31, 20 2025.**.

During this internship, he demonstrated exceptional dedication, was expo processes, and was found diligent, hardworking, and committed to his contributing positively to the End-to-End Development of Project Titled **"AI-ba Well-AI"** using React.js, Tailwind CSS for Frontend Development and Py Models for Backend Development under the **KAVIGAI Internship Program.**

We extend our heartfelt congratulations on his successful completion of the int and wish him all the best in his future endeavors.

With Best Regards,

Suganya - M

Suganya Munuswamy,
Mobile App Developer/Project Coordinator,
NASO Technologies Pvt. Ltd.
Chennai, India
Mobile# +91-8939738647

Date: 28 - 04 - 2025