Back End Engineering-I

Project Report

Semester-IV (Batch-2023)

Group Of Institutions

Supervised By:

Mr. Rohit Kumar

Faculty Name:

Dr. Gurtej Kaur

Submitted By:

Ishaan Rai 23109911933 G22

Harshit Gupta 2310991927 G22

Swapnil Gaur 2310991222 G22

Department of Computer Science and Engineering
Chitkara University Institute of Engineering & Technology,
Chitkara University, Punjab

ABSTRACT

Mental health has emerged as one of the most pressing concerns in today’s fast-paced world, with millions of people struggling with stress, anxiety, and depression. Despite the growing need, professional therapy and counselling often remain inaccessible due to high costs, social stigma, and limited availability of mental health professionals. To address this gap, **NeuroSync – AI Mental Health Companion** has been developed as an innovative platform that combines advanced artificial intelligence with modern web technologies to provide continuous, personalized mental health support.

NeuroSync offers a wide range of features, including an **AI-powered therapy chat** driven by LLAMA 3, **real-time emotion analysis**, **24/7 crisis support**, and **personalized guidance** for users. The platform also incorporates **progress tracking**, enabling individuals to monitor their emotional growth over time. Built using **Next.js, React, and Tailwind CSS** on the frontend, and **MongoDB with JWT-based authentication** on the backend, NeuroSync ensures a secure, scalable, and user-friendly experience.

A key strength of NeuroSync lies in its ability to integrate AI-driven emotion detection with conversational support, allowing users to receive empathetic, non-judgmental guidance anytime. By leveraging **custom emotion analysis algorithms** and **real-time mood detection**, the system bridges the gap between technology and mental health care. Moreover, strong emphasis is placed on **data security and privacy**, with all sensitive information stored securely using modern encryption standards.

The project not only demonstrates the potential of AI in healthcare but also provides a foundation for future enhancements, such as integration with wearable devices, multilingual support, mobile applications, and hybrid therapy models that connect users with certified professionals. NeuroSync ultimately envisions a future where technology plays a central role in making mental health care more accessible, affordable, and stigma-free for individuals worldwide.

Table Of Contents

1. INTRODUCTION

1.1 Background:

Mental health is an essential component of overall well-being, yet it continues to be one of the most neglected areas in healthcare. With rising cases of anxiety, depression, and stress disorders, there is a growing demand for accessible mental health solutions. Traditional therapy, though effective, is often hindered by high costs, stigma, and a shortage of trained professionals. Additionally, most individuals lack tools to continuously monitor their emotional well-being. The advancement of Artificial Intelligence (AI) and digital health technologies opens up new opportunities to bridge this gap by offering scalable, affordable, and personalized mental health support. NeuroSync has been conceptualized to leverage these technologies to provide an innovative AI-powered mental health companion.

1.2 Objectives:

- 🤖 To develop an AI-driven therapy chat system capable of empathetic and human-like conversations.
- 📊 To implement real-time emotion analysis and mood detection for personalized support.
- 🆘 To provide 24/7 crisis intervention and immediate assistance in high-risk situations.
- 🎯 To deliver tailored mental health guidance and coping strategies for individual needs.
- 📈 To enable users to track their emotional growth and progress over time.
- 🔒 To ensure secure authentication and protect sensitive user data through modern encryption methods.
- 🌍 To create a scalable, user-friendly platform accessible across devices.

1.3 Significance:

The significance of NeuroSync lies in its potential to revolutionize the way individuals access mental health care. By integrating conversational AI, emotion analysis, and secure digital platforms, it offers:

- Accessibility – Support available anytime, anywhere, without geographical or financial barriers.
- Affordability – Cost-effective alternative to traditional therapy sessions.
- Personalization – Tailored mental health guidance based on individual emotions and needs.
- Crisis Readiness – Immediate support during emergencies, reducing risks of harm.
- Progress Awareness – Continuous tracking of emotional health for self-reflection and growth.
- Technological Advancement – Demonstrates the role of AI in addressing global healthcare challenges.

Ultimately, NeuroSync serves as a step forward in making mental health support more inclusive, scalable, and stigma-free, while also showcasing how AI can be harnessed for social good.

2. PROBLEM DEFINITION AND REQUIREMENTS:

2.1 Problem Statement:

Mental health issues such as stress, anxiety, and depression are rapidly increasing worldwide, yet access to timely professional help remains limited. Traditional therapy often faces barriers including:

- High cost of counselling sessions.
- Limited availability of certified mental health professionals.
- Social stigma associated with seeking therapy.
- Lack of real-time monitoring and intervention.

Many individuals do not have access to continuous support, personalized guidance, or reliable tools to track their emotional well-being. This highlights the urgent need for a secure, AI-powered platform that can provide **24/7 assistance, emotional analysis, and crisis intervention** while ensuring accessibility, affordability, and privacy.

2.2 <u>Software Requirements:</u>

<u>Frontend</u>

- Next.js – React framework for building the user interface and routing
- React.js – Component-based UI development
- Tailwind CSS – Responsive and modern styling
- npm – Package manager for frontend dependencies

<u>Backend</u>

- Node.js – Server-side runtime environment
- Next.js API Routes / Express.js – API handling
- MongoDB – Database for storing user details and progress tracking
- Mongoose – ODM for MongoDB schema management

<u>Authentication & Security</u>

- JWT (JSON Web Tokens) – User authentication and session handling
- bcrypt – Secure password hashing
- dotenv – Managing environment variables securely

<u>AI Integration</u>

- LLAMA 3 Model – Conversational AI engine
- Custom Emotion Analysis Algorithms – For mood detection and emotional insights

<u>Other Tools</u>

- Git & GitHub – Version control
- Vercel/Netlify – Deployment
- Postman / Thunder Client – API testing

2.3 Hardware Requirements:

Minimum System Requirements (for Development)

- Processor: Intel i3 (8th Gen) / AMD Ryzen 3 or higher
- RAM: 8 GB
- Storage: 256 GB SSD (or HDD with more space)
- Operating System: Windows 10 / Linux / macOS
- Internet: Stable broadband connection for API and DB connectivity

Recommended System Requirements (for Smooth Development & Deployment)

- Processor: Intel i5/i7 (10th Gen) / AMD Ryzen 5 or higher
- RAM: 16 GB
- Storage: 512 GB SSD
- GPU (Optional): NVIDIA GTX/RTX series for faster AI model integration
- Operating System: Latest Windows 11 / Ubuntu 22.04 / macOS Monterey or higher

Server/Deployment Requirements

- Cloud Hosting: AWS / GCP / Azure or Vercel for deployment
- Database Hosting: MongoDB Atlas or self-hosted MongoDB server
- Node.js Runtime: v16 or higher
- RAM: 4 GB+ (for production server

3. PROPOSED DESIGN / METHODOLOGY

3.1 System Overview:

NeuroSync – AI Mental Health Companion is designed as a web-based platform that integrates conversational AI, emotion detection, and secure user management to provide continuous mental health support. The system consists of a frontend application, a backend server, an AI module, and a database layer.

- The frontend (Next.js + Tailwind CSS) provides users with a responsive, intuitive interface that supports real-time chat, progress tracking, and dark mode.
- The backend (Node.js + MongoDB) manages authentication, data storage, and communication between the frontend and AI models.
- The AI engine (LLAMA 3 + custom emotion analysis algorithms) powers therapy-like conversations, real-time sentiment detection, and mood tracking.
- The system ensures security through JWT-based authentication, bcrypt password hashing, and secure environment variable management.

3.2 Design Methodology:

The project follows a **modular design methodology**, ensuring scalability, maintainability, and user security:

1. **Requirement Analysis**
    - Identify core mental health challenges and user needs (therapy chat, emotion analysis, crisis support).
2. **System Design**
    - Architecture split into **Frontend, Backend, AI, and Database** layers.
    - Emphasis on security and user privacy.
3. **Development Phase**
    - **Frontend**: Next.js + Tailwind CSS for a responsive, modern UI.
    - **Backend**: Node.js APIs for authentication, chat handling, and progress tracking.
    - **AI Integration**: LLAMA 3 for therapy chat and custom emotion analysis.
4. **Testing**
    - Functional testing for chat, login, and data tracking.
    - Security testing for authentication and data protection.
5. **Deployment**
    - Deployment on **Vercel / Cloud Hosting**.
    - Database hosted on **MongoDB Atlas** for scalability.
6. **Maintenance & Enhancements**
    - Regular updates for model improvement, new features, and security patches.

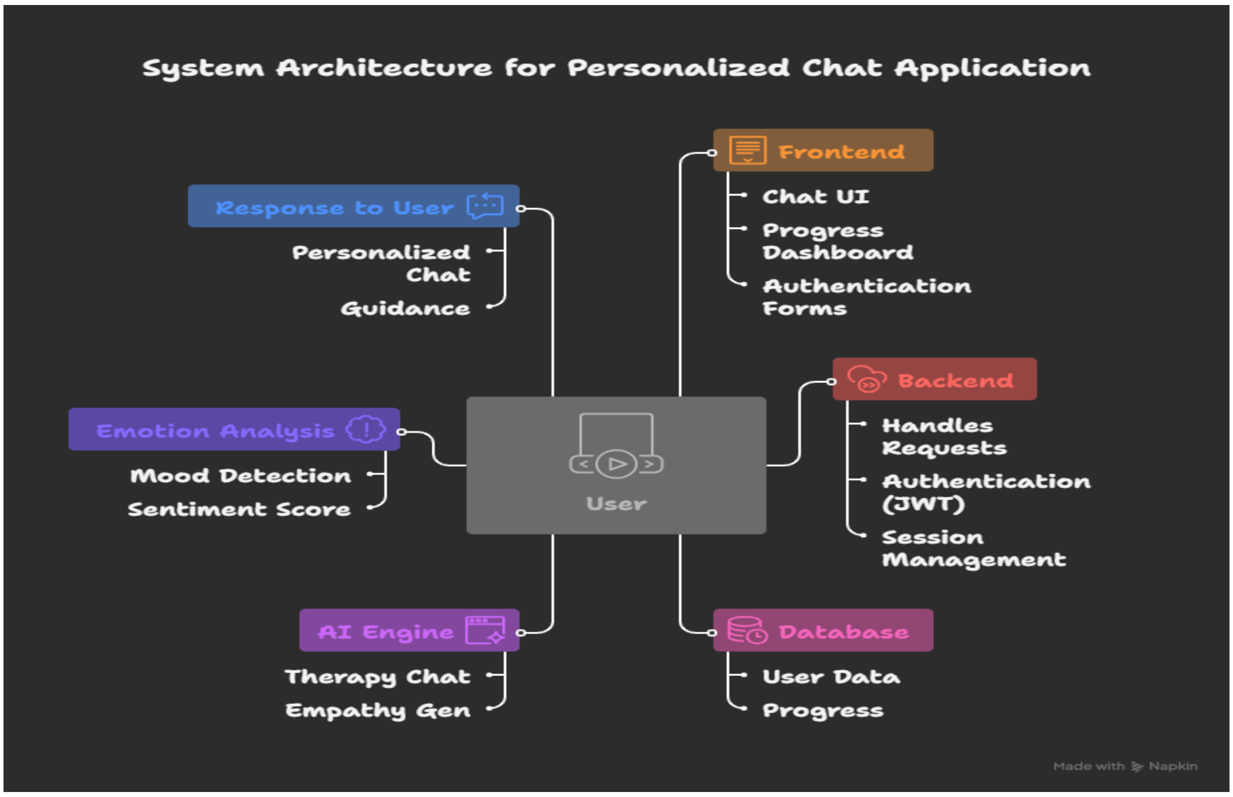
3.3 Architecture / File Structure:

```
🗀 Backend Project/
├── 🖹 .env                 # Environment variables (not tracked)
├── 🖹 .env.example             # Environment variables template
├── 🖹 .gitignore              # Git ignore rules
├── 🖹 LICENSE                 # MIT License
├── 🖹 README.md                 # Project documentation
├── 🖹 SETUP.md                # Setup instructions
├── 🖹 package.json             # Node.js dependencies & scripts
├── 🖹 package-lock.json          # Dependency lock file
├── 🖹 next.config.js            # Next.js configuration
├── 🖹 postcss.config.js          # PostCSS configuration
├── 🖹 tailwind.config.js          # TailwindCSS configuration
│
├── 🗀 app/                  # Alternative app structure (unused)
│   ├── 🗀 components/
│   │   ├── 🖹 Footer.jsx
│   │   ├── 🖹 Layout.jsx
│   │   └── 🖹 Navbar.jsx
│   └── 🗀 styles/
│       └── 🖹 globals.css
│
```

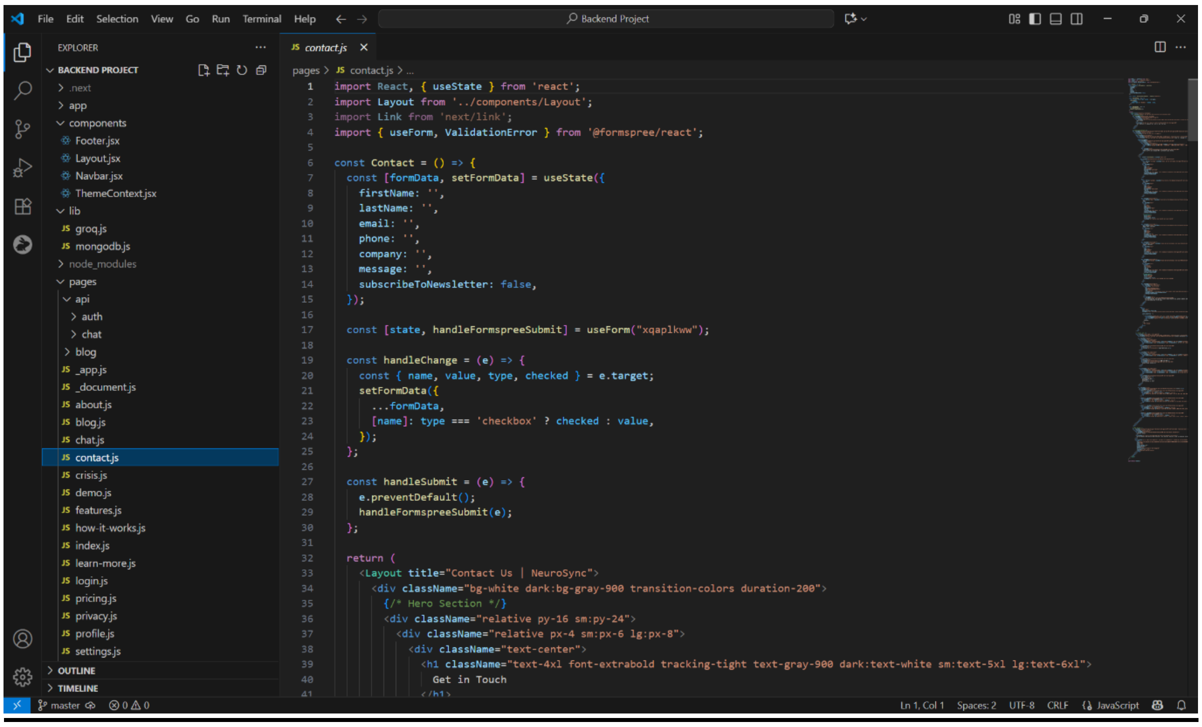
```
├── 🗀 components/           # Reusable React components
│   ├── 🖹 Footer.jsx          # Footer component
│   ├── 🖹 Layout.jsx          # Main layout wrapper
│   ├── 🖹 Navbar.jsx          # Navigation bar
│   └── 🖹 ThemeContext.jsx    # Theme context provider
│
├── 🗀 lib/                  # Utility libraries
│   ├── 🖹 groq.js             # Groq AI integration & response cleaning
│   └── 🖹 mongodb.js          # MongoDB connection & utilities
│
├── 🗀 pages/                # Next.js pages & API routes
│   ├── 🖹 _app.js             # Custom App component
│   ├── 🖹 _document.js        # Custom Document component
│   ├── 🖹 about.js            # About page
│   ├── 🖹 blog.js             # Blog listing page
│   ├── 🖹 chat.js             # Main chat interface
│   ├── 🖹 contact.js          # Contact page
│   ├── 🖹 crisis.js           # Crisis support page
│   ├── 🖹 demo.js             # Demo chat page
│   ├── 🖹 features.js         # Features page
│   ├── 🖹 how-it-works.js     # How it works page
│   ├── 🖹 index.js            # Homepage
│   ├── 🖹 learn-more.js       # Learn more page
│   ├── 📄 login.js          # User login page
│   ├── 📄 pricing.js        # Pricing page
│   ├── 📄 privacy.js        # Privacy policy page
│   ├── 📄 profile.js        # User profile page
│   ├── 📄 settings.js       # User settings page
│   ├── 📄 signup.js         # User registration page
│   ├── 📄 terms.js          # Terms of service page
│   │
│   ├── 📁 api/              # Backend API endpoints
│   │   ├── 📁 auth/         # Authentication endpoints
│   │   │   ├── 📄 login.js      # POST - User login
│   │   │   ├── 📄 register.js   # POST - User registration
│   │   │   └── 📄 update-password.js # POST - Password update
│   │   │
│   │   └── 📁 chat/         # Chat-related endpoints
│   │       ├── 📄 history.js    # GET - Chat history
│   │       ├── 📄 message.js    # POST - Send message & get AI response
│   │       │
│   │       └── 📁 conversation/
│   │           └── 📄 [id].js   # GET - Specific conversation
│   │
│   └── 📁 blog/
│       └── 📄 [id].js       # Dynamic blog post pages
```

```
├── 🗀 public/               # Static assets
│   └── 🗀 images/
│       └── 🖹 neural-network.webp   # Hero image
│
├── 🗀 styles/               # Global styles
│   └── 🖹 globals.css          # Global CSS with Tailwind imports
│
├── 🗀 .git/              # Git repository data
├── 🗀 .next/              # Next.js build cache & output
└── 🗀 node_modules/          # NPM dependencies
```

3.4 Systematic Diagram:

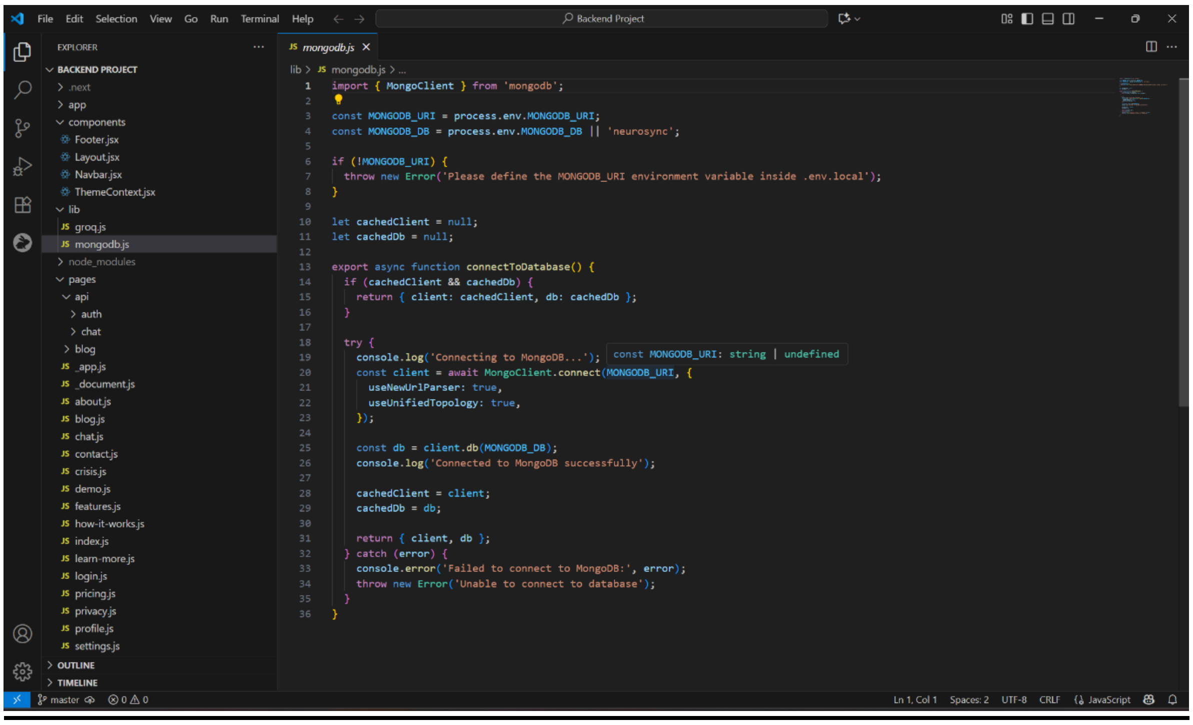


4. Results (Code Snippets and the Project Snapshots)



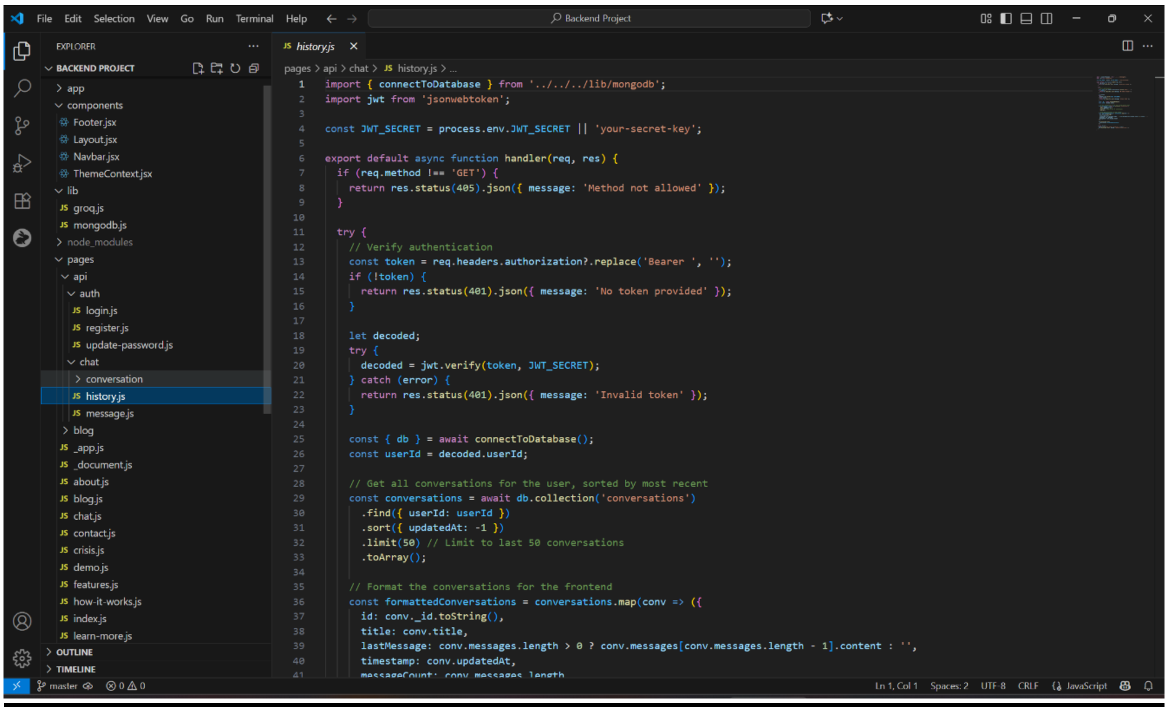
```javascript
import React, { useState } from 'react';
import Layout from '../components/Layout';
import Link from 'next/link';
import { useForm, ValidationError } from '@formspree/react';

const Contact = () => {
  const [formData, setFormData] = useState({
    firstName: '',
    lastName: '',
    email: '',
    phone: '',
    company: '',
    message: '',
    subscribeToNewsletter: false,
  });

  const [state, handleFormspreeSubmit] = useForm("xqaplkww");

  const handleChange = (e) => {
    const { name, value, type, checked } = e.target;
    setFormData({
      ...formData,
      [name]: type === 'checkbox' ? checked : value,
    });
  };

  const handleSubmit = (e) => {
    e.preventDefault();
    handleFormspreeSubmit(e);
  };

  return (
    <Layout title="Contact Us | NeuroSync">
      <div className="bg-white dark:bg-gray-900 transition-colors duration-200">
        {/* Hero Section */}
        <div className="relative py-16 sm:py-24">
          <div className="relative px-4 sm:px-6 lg:px-8">
            <div className="text-center">
              <h1 className="text-4xl font-extrabold tracking-tight text-gray-900 dark:text-white sm:text-5xl lg:text-6xl">
                Get in Touch
              </h1>
```

Figure 1Contact.js



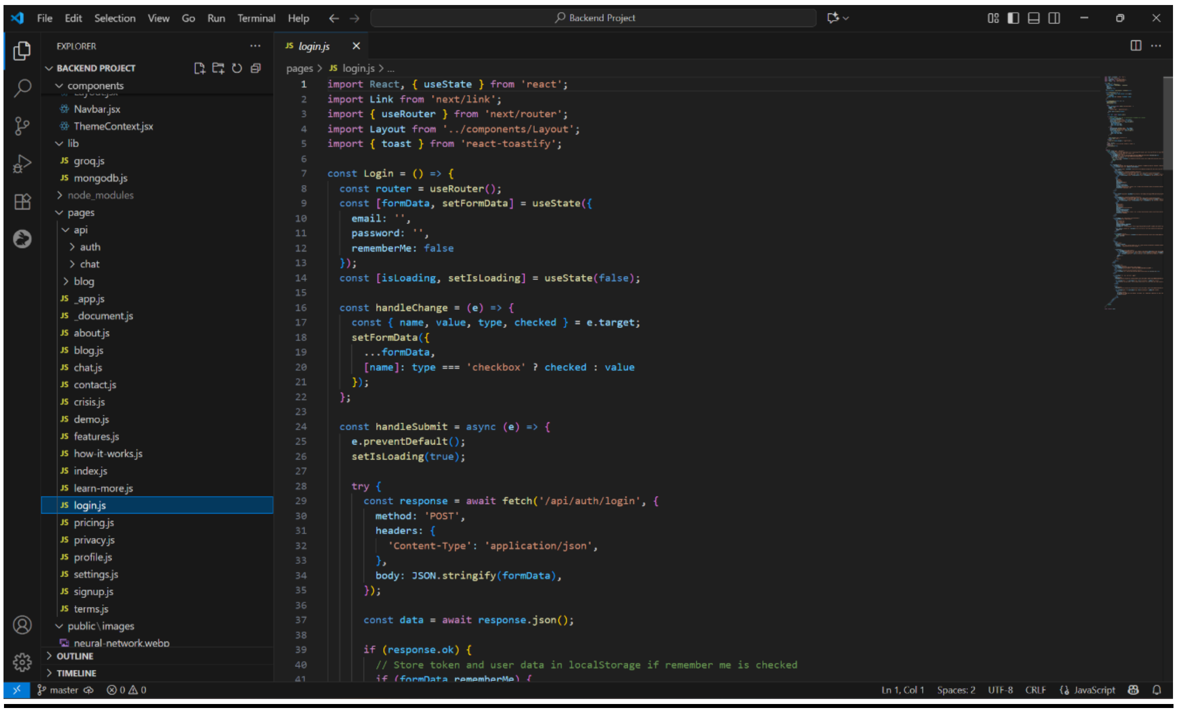
```javascript
import { MongoClient } from 'mongodb';

const MONGODB_URI = process.env.MONGODB_URI;
const MONGODB_DB = process.env.MONGODB_DB || 'neurosync';

if (!MONGODB_URI) {
  throw new Error('Please define the MONGODB_URI environment variable inside .env.local');
}

let cachedClient = null;
let cachedDb = null;

export async function connectToDatabase() {
  if (cachedClient && cachedDb) {
    return { client: cachedClient, db: cachedDb };
  }

  try {
    console.log('Connecting to MongoDB...');
    const client = await MongoClient.connect(MONGODB_URI, {
      useNewUrlParser: true,
      useUnifiedTopology: true,
    });

    const db = client.db(MONGODB_DB);
    console.log('Connected to MongoDB successfully');

    cachedClient = client;
    cachedDb = db;

    return { client, db };
  } catch (error) {
    console.error('Failed to connect to MongoDB:', error);
    throw new Error('Unable to connect to database');
  }
}
```

Figure 2MongoDB.js



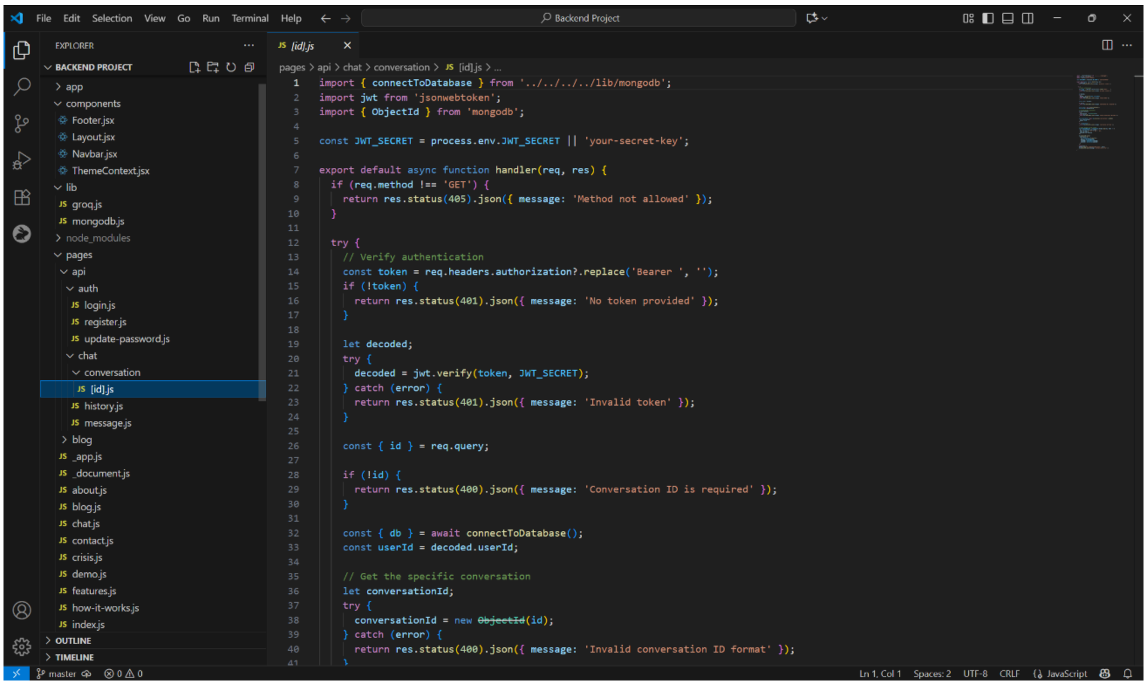
```javascript
import { connectToDatabase } from '../../../lib/mongodb';
import jwt from 'jsonwebtoken';

const JWT_SECRET = process.env.JWT_SECRET || 'your-secret-key';

export default async function handler(req, res) {
  if (req.method !== 'GET') {
    return res.status(405).json({ message: 'Method not allowed' });
  }

  try {
    // Verify authentication
    const token = req.headers.authorization?.replace('Bearer ', '');
    if (!token) {
      return res.status(401).json({ message: 'No token provided' });
    }

    let decoded;
    try {
      decoded = jwt.verify(token, JWT_SECRET);
    } catch (error) {
      return res.status(401).json({ message: 'Invalid token' });
    }

    const { db } = await connectToDatabase();
    const userId = decoded.userId;

    // Get all conversations for the user, sorted by most recent
    const conversations = await db.collection('conversations')
      .find({ userId: userId })
      .sort({ updatedAt: -1 })
      .limit(50) // Limit to last 50 conversations
      .toArray();

    // Format the conversations for the frontend
    const formattedConversations = conversations.map(conv => ({
      id: conv._id.toString(),
      title: conv.title,
      lastMessage: conv.messages.length > 0 ? conv.messages[conv.messages.length - 1].content : '',
      timestamp: conv.updatedAt,
      messageCount: conv.messages.length
```

Figure 3History.js



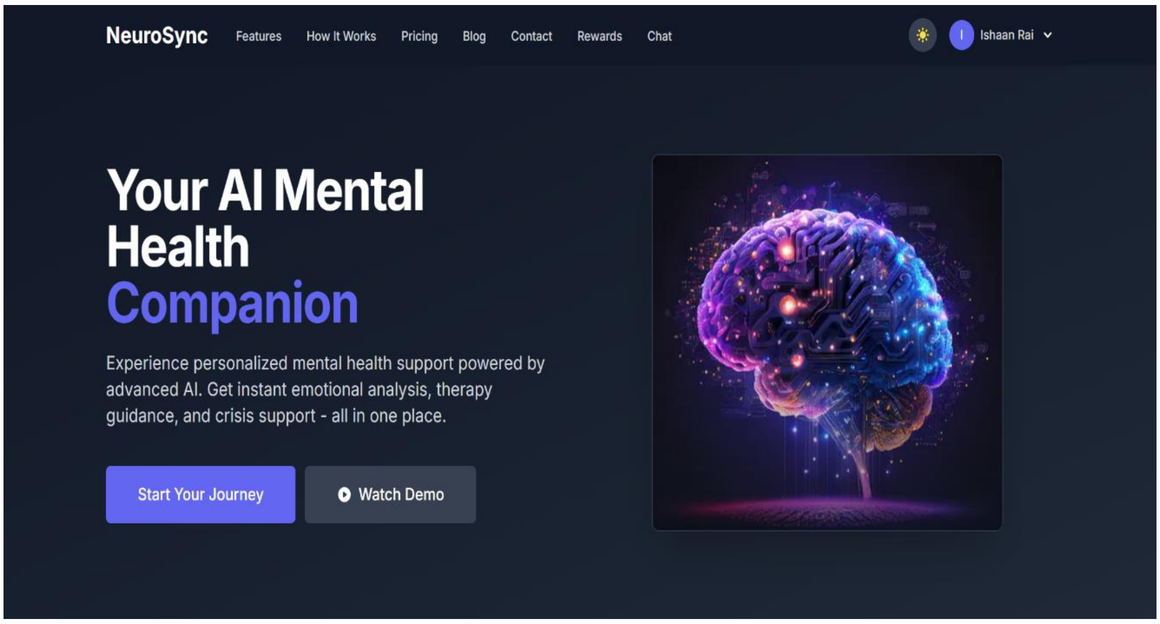
```javascript
import React, { useState } from 'react';
import Link from 'next/link';
import { useRouter } from 'next/router';
import Layout from '../components/Layout';
import { toast } from 'react-toastify';

const Login = () => {
  const router = useRouter();
  const [formData, setFormData] = useState({
    email: '',
    password: '',
    rememberMe: false
  });
  const [isLoading, setIsLoading] = useState(false);

  const handleChange = (e) => {
    const { name, value, type, checked } = e.target;
    setFormData({
      ...formData,
      [name]: type === 'checkbox' ? checked : value
    });
  };

  const handleSubmit = async (e) => {
    e.preventDefault();
    setIsLoading(true);

    try {
      const response = await fetch('/api/auth/login', {
        method: 'POST',
        headers: {
          'Content-Type': 'application/json',
        },
        body: JSON.stringify(formData),
      });

      const data = await response.json();

      if (response.ok) {
        // Store token and user data in localStorage if remember me is checked
        if (formData.rememberMe) {
```

Figure 4Login.js



```javascript
import { connectToDatabase } from '../../../../lib/mongodb';
import jwt from 'jsonwebtoken';
import { ObjectId } from 'mongodb';

const JWT_SECRET = process.env.JWT_SECRET || 'your-secret-key';

export default async function handler(req, res) {
  if (req.method !== 'GET') {
    return res.status(405).json({ message: 'Method not allowed' });
  }

  try {
    // Verify authentication
    const token = req.headers.authorization?.replace('Bearer ', '');
    if (!token) {
      return res.status(401).json({ message: 'No token provided' });
    }

    let decoded;
    try {
      decoded = jwt.verify(token, JWT_SECRET);
    } catch (error) {
      return res.status(401).json({ message: 'Invalid token' });
    }

    const { id } = req.query;

    if (!id) {
      return res.status(400).json({ message: 'Conversation ID is required' });
    }

    const { db } = await connectToDatabase();
    const userId = decoded.userId;

    // Get the specific conversation
    let conversationId;
    try {
      conversationId = new ~~ObjectId~~(id);
    } catch (error) {
      return res.status(400).json({ message: 'Invalid conversation ID format' });
    }
```

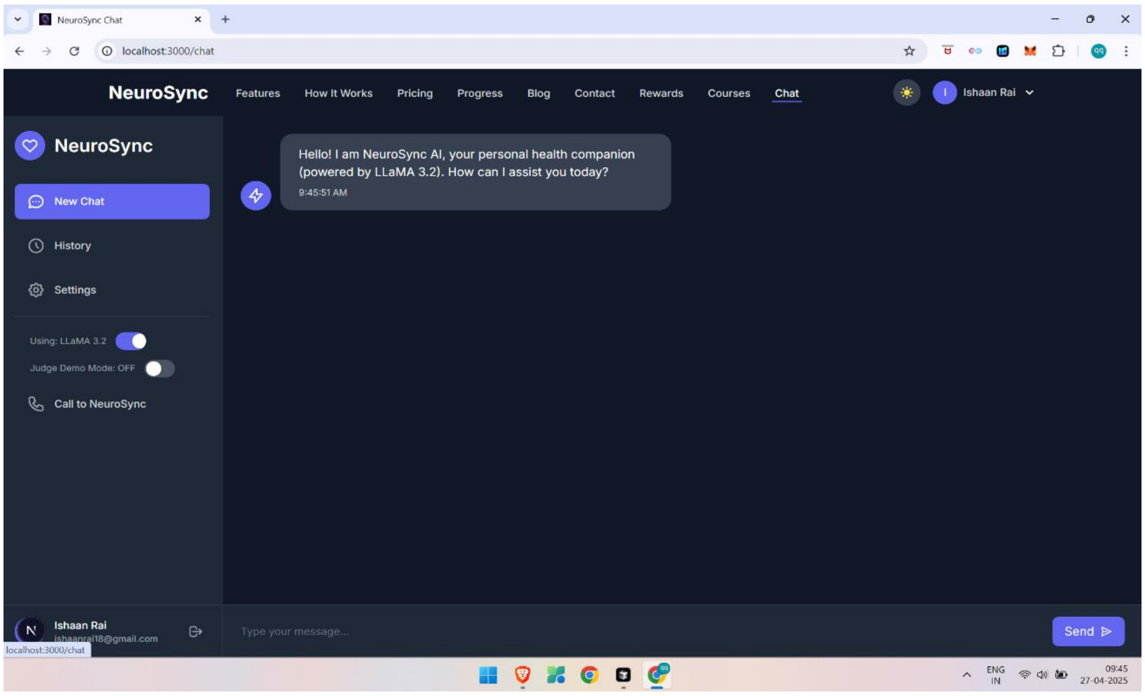
Figure 5Conversations.js



Your AI Mental Health Companion

Experience personalized mental health support powered by advanced AI. Get instant emotional analysis, therapy guidance, and crisis support - all in one place.

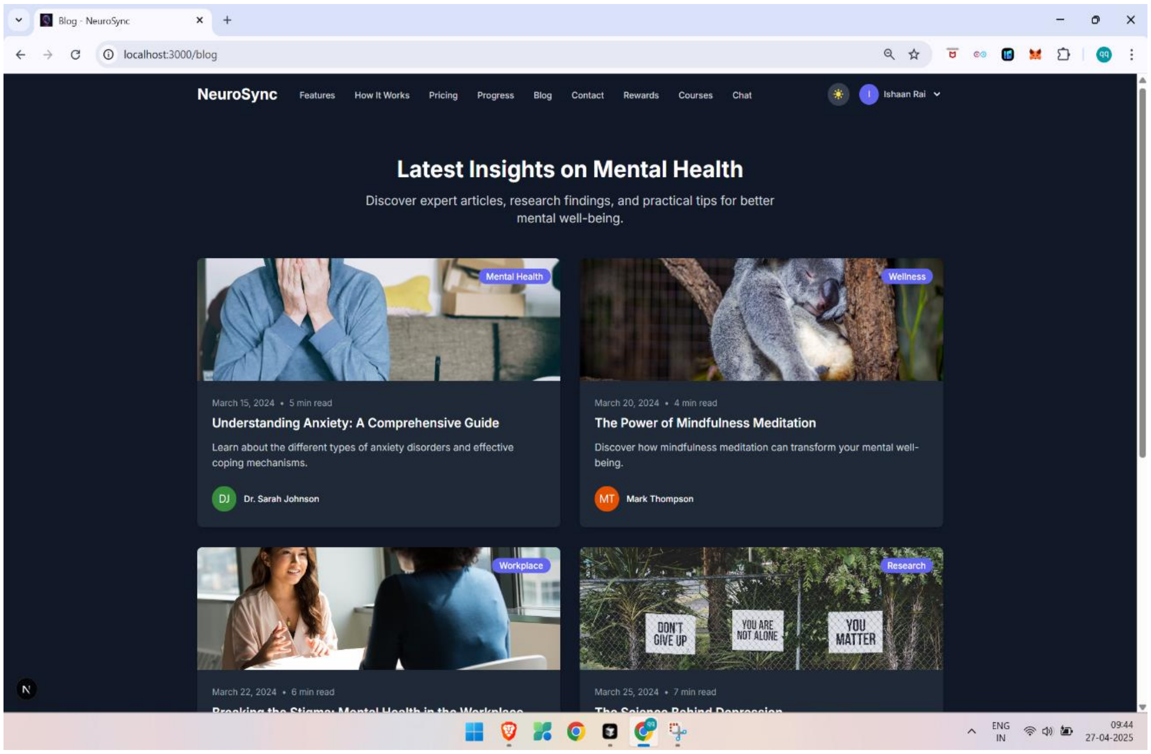
Figure 6Home Page



Hello! I am NeuroSync AI, your personal health companion (powered by LLaMA 3.2). How can I assist you today?
9:45:51 AM

Ishaan Rai
ishaanrai18@gmail.com

Figure 7 Chat Page



Latest Insights on Mental Health

Discover expert articles, research findings, and practical tips for better mental well-being.

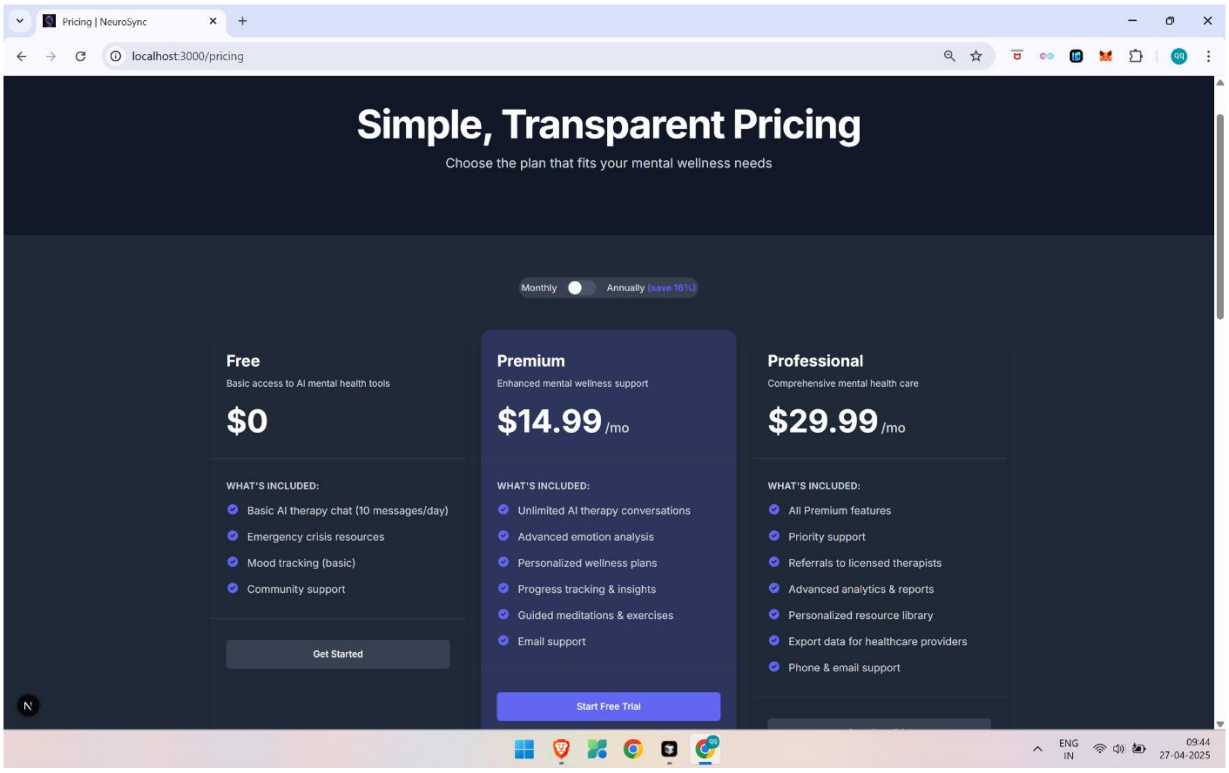
March 15, 2024 • 5 min read
Understanding Anxiety: A Comprehensive Guide
Learn about the different types of anxiety disorders and effective coping mechanisms.
DJ Dr. Sarah Johnson

March 20, 2024 • 4 min read
The Power of Mindfulness Meditation
Discover how mindfulness meditation can transform your mental well-being.
MT Mark Thompson

March 22, 2024 • 6 min read
Breaking the Stigma: Mental Health in the Workplace

March 25, 2024 • 7 min read
The Science Behind Depression

Figure 8 Blog Page



Simple, Transparent Pricing

Choose the plan that fits your mental wellness needs

Monthly Annually (save 16%)

Free
Basic access to AI mental health tools
$0

WHAT'S INCLUDED:
- Basic AI therapy chat (10 messages/day)
- Emergency crisis resources
- Mood tracking (basic)
- Community support

Get Started

Premium
Enhanced mental wellness support
$14.99 /mo

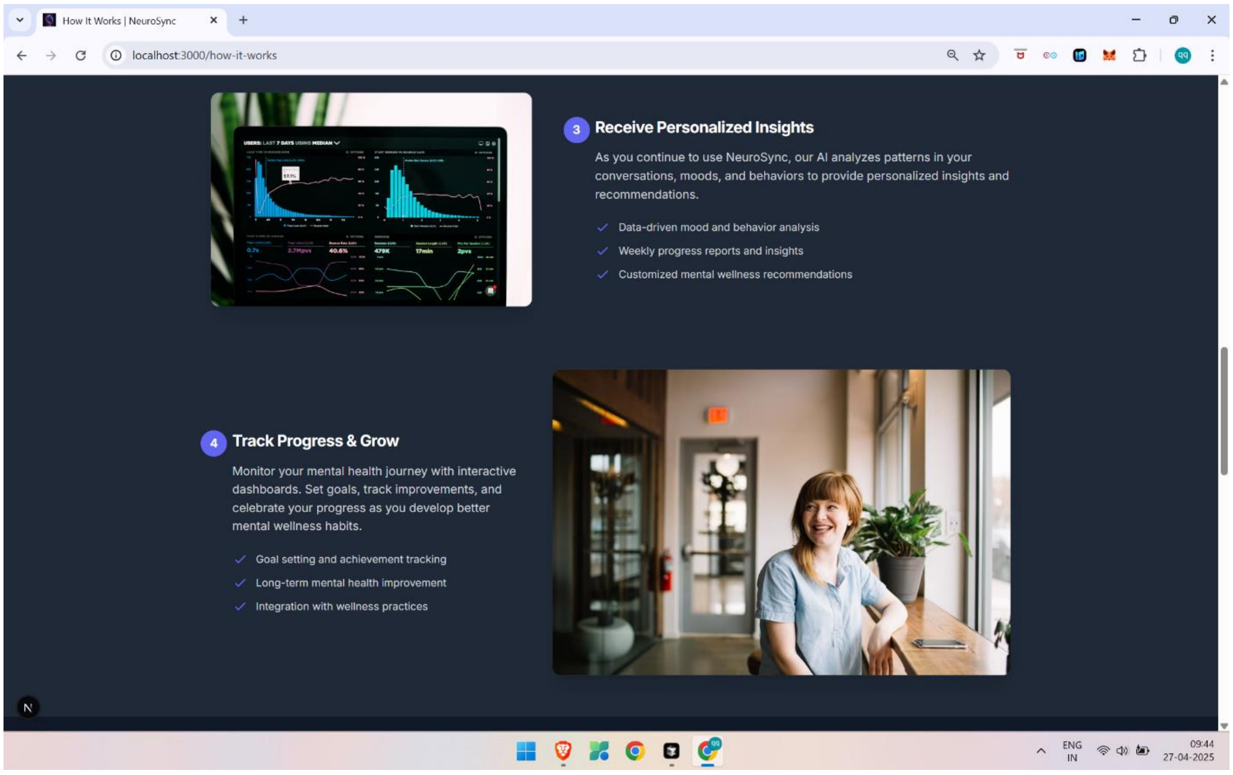
WHAT'S INCLUDED:
- Unlimited AI therapy conversations
- Advanced emotion analysis
- Personalized wellness plans
- Progress tracking & insights
- Guided meditations & exercises
- Email support

Start Free Trial

Professional
Comprehensive mental health care
$29.99 /mo

WHAT'S INCLUDED:
- All Premium features
- Priority support
- Referrals to licensed therapists
- Advanced analytics & reports
- Personalized resource library
- Export data for healthcare providers
- Phone & email support

Figure 9 Pricing Page



3 **Receive Personalized Insights**

As you continue to use NeuroSync, our AI analyzes patterns in your conversations, moods, and behaviors to provide personalized insights and recommendations.

- ✓ Data-driven mood and behavior analysis
- ✓ Weekly progress reports and insights
- ✓ Customized mental wellness recommendations

4 **Track Progress & Grow**

Monitor your mental health journey with interactive dashboards. Set goals, track improvements, and celebrate your progress as you develop better mental wellness habits.

- ✓ Goal setting and achievement tracking
- ✓ Long-term mental health improvement
- ✓ Integration with wellness practices

Figure 10 Features Page

5. REFERENCES

- **Node.js Documentation:** https://nodejs.org
- **Express.js Documentation:** https://expressjs.com
- **MongoDB Docs:** https://docs.mongodb.com
- **Postman Docs:** https://learning.postman.com
- **Mongoose Docs:** https://mongoosejs.com