

Real Time Application to Manage Farms More Efficiently

A Mini Project Report

Submitted in partial fulfilment of the
Requirements for the award of the Degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE & ENGINEERING

By

Mani Kumar Dodla (1602-22-748-019)

Sree Vaibhav Duvvuri (1602-22-748-044)

Under the guidance of

Dr R Sateesh Kumar

Assistant Professor



Department of Computer Science & Engineering

Vasavi College of Engineering (Autonomous)

(Affiliated to Osmania University)

Ibrahimbagh, Hyderabad-31

2024

Vasavi College of Engineering (Autonomous)
(Affiliated to Osmania University) Hyderabad- 500 031

Department of Computer Science & Engineering



DECLARATION BY THE CANDIDATE

We, **Mani Kumar Dodla, Sree Vaibhav Duvvuri** bearing hall ticket numbers **1602-22-748-019, 1602-22-748-044** hereby declare that the paper publication titled **“Real Time Application to Manage Farms More Efficiently”** under the guidance of **Dr R Sateesh Kumar**, Assistant Professor, Department of Computer Science & Engineering, VCE, Hyderabad, is submitted in partial fulfilment of the requirement for the award of the degree of **Bachelor of Engineering in Computer Science & Engineering**.

This is a record of bonafide work carried out by me and the results embodied in this project report have not been submitted to any other university or institute for the award of any other degree or diploma.

Mani Kumar Dodla (1602-22-748-019)
Sree Vaibhav Duvvuri (1602-22-748-044)

Vasavi College of Engineering (Autonomous)
(Affiliated to Osmania University) Hyderabad-500 031

Department of Computer Science & Engineering



BONAFIDE CERTIFICATE

This is to certify that the project entitled
“Real Time Application to Manage Farms More Efficiently” being submitted by
Mani Kumar Dodla and Sree Vaibhav Duvvuri bearing **1602-22-748-019 and 1602-22-748-044** in partial fulfilment of the requirements for the award of the degree of Bachelor of Engineering in Computer Science & Engineering is a record of bonafide work carried out by him/her under my guidance.

Dr R Sateesh Kumar,
Assistant Professor,
Internal Guide

Dr. T. Adilakshmi,
Professor &HOD,
Dept. of CSE.

ACKNOWLEDGEMENT

We take this opportunity with pride and enormous gratitude, to express the deeply embedded feeling and gratefulness to our respectable guide **Dr R Sateesh Kumar**, Department of Computer Science and Engineering, whose guidance was unforgettable and filled with innovative ideas as well as her constructive suggestions has made the presentation of my major project a grand success.

We are thankful to **Dr. T. Adilakshmi**, Head of Department (CSE), **Vasavi College of Engineering** for the help during our course work.

Finally we express our gratitude to the management of our college, **Vasavi College of Engineering** for providing the necessary arrangements and support to complete my paper work successfully.

ABSTRACT

Abstract

AgriGuard is an innovative web application designed to transform agricultural practices by leveraging NASA's real-time data and cutting-edge artificial intelligence. The platform empowers farmers with smart crop management tools to enhance productivity, conserve water, and adapt to changing environmental conditions. By integrating advanced technologies, AgriGuard addresses critical challenges in agriculture, such as optimizing irrigation, mitigating flood risks, and making data-driven crop choices.

The application features an AI-driven crop recommendation system that utilizes real-time precipitation and groundwater data from the NASA POWER API. This ensures farmers can select crops best suited to their local environmental conditions. Additionally, AgriGuard provides irrigation guidance by analyzing rainfall patterns and soil moisture levels, helping farmers optimize water usage. The inclusion of flood alerts, powered by Open Meteo river discharge data, ensures proactive disaster preparedness, reducing potential crop losses.

AgriGuard also emphasizes user-centric design with visually rich data representations through Recharts, an intuitive AI chatbot powered by Hugging Face Mistral AI for immediate support, and direct access to agricultural experts for tailored advice. A dedicated section on water conservation and crop-specific water requirements further supports sustainable farming practices.

Built with React and Tailwind CSS for a responsive frontend, AgriGuard combines open data from NASA, Open Meteo, and OpenWeather with advanced AI to create a comprehensive, accessible, and practical tool for modern agriculture. Future iterations aim to expand real-time data integration and enhance user engagement, reinforcing AgriGuard's mission to foster sustainable and resilient farming worldwide.

TABLE OF CONTENTS

1. INTRODUCTION
 - i. OVERVIEW
 - ii. PROJECT SCOPE
2. SYSTEM REQUIREMENTS
 - i. HARDWARE REQUIREMENTS
 - ii. SOFTWARE REQUIREMENTS
3. IMPLEMENTATION
 - i. main.jsx
 - ii. App.jsx
 - iii. Homepage.jsx
 - iv. About.jsx
4. RESULTS
 - i. Web Start Interface.jpeg
 - ii. Homepage.jpeg
 - iii. Chatbot.jpeg
 - iv. About.jpeg
5. CONCLUSION
6. REFERENCES

1. INTRODUCTION

1.1 Overview

The **AgriGuard Platform** is a web-based application designed to revolutionize agriculture by leveraging real-time data and artificial intelligence. The platform integrates advanced tools such as AI-assisted crop recommendations, irrigation guidance, and real-time flood alerts, offering a streamlined and effective solution for modern farming.

Built with a front-end developed in React and Tailwind CSS, and a back-end powered by APIs like NASA POWER, Open Meteo, and OpenWeather, AgriGuard ensures seamless user experiences while delivering precise agricultural insights. The platform provides real-time data visualization, AI-driven support, and expert consultations, making it suitable for farmers, agricultural professionals, and organizations alike.

By combining AI-powered tools, real-time data, and user-friendly features, AgriGuard addresses key challenges in agriculture. It empowers users to make informed decisions about crop selection, water usage, and disaster preparedness, fostering productivity and sustainability in farming practices.

1.2 Project Scope

AgriGuard enhances agricultural management through the following core components:

1. **Crop Recommendations:**
 - Uses NASA POWER API for real-time precipitation and groundwater data.
 - Suggests optimal crops tailored to local environmental conditions.
2. **Irrigation Guidance:**
 - Analyzes rainfall patterns to provide water usage recommendations.
 - Promotes efficient irrigation and water conservation.
3. **Flood Alerts:**
 - Integrates Open Meteo river discharge data for early flood warnings.
 - Mitigates risks by enabling proactive disaster preparedness.
4. **User Interface:**
 - Offers a responsive, minimalistic design with Recharts for data visualization.
 - Simplifies access to insights and actionable recommendations.
5. **AI Assistance:**
 - Provides instant support with an AI-powered chatbot for queries.
 - Suggests water-saving tips and answers common agricultural concerns.
6. **Back-End Functionality:**
 - Powered by APIs for secure, real-time data updates.
 - Supports user authentication and seamless integration of third-party tools.
7. **Potential for Expansion:**
 - Modular design for future additions like live expert chats, video support, and more data sources.
 - Adaptable for educational use and diverse agricultural needs.

This scalable platform offers farmers an efficient, accessible, and data-driven approach to sustainable agriculture, addressing diverse user requirements.

2. System Requirements

2.1 Hardware Requirements

To efficiently run the AgriGuard platform, the following hardware specifications are recommended:

- **Processor:** Intel Core i3 or equivalent
- **RAM:** Minimum 4 GB (8 GB recommended for optimal performance)
- **Storage:** At least 20 GB of free disk space for data storage
- **Network:** A stable internet connection with a minimum speed of 5 Mbps for real-time updates
- **Operating System:** Windows 10/11, macOS, or Linux (Ubuntu 18.04 or higher)

2.2 Software Requirements

- **Operating System:** Windows 10, macOS, or Linux-based OS (Ubuntu 18.04+)
- **Programming Language:** Node.js 16.x+
- **Frameworks/Libraries:** React, Tailwind CSS, NASA POWER API, Open Meteo API, OpenWeather API
- **Database:** Firebase (real-time database for data synchronization and storage)
- **Additional Tools:** npm (for dependency management), Environment Variables (for API keys), Postman (for API testing)

These specifications ensure smooth functionality and scalability, enabling AgriGuard to cater to a wide range of users and agricultural scenarios.

3. IMPLEMENTATION

main.jsx

```
import { StrictMode } from 'react'
import { createRoot } from 'react-dom/client'
import { createBrowserRouter, RouterProvider } from 'react-router-dom'
import App from './App.jsx'
import './index.css'
import Homepage from './pages/Homepage.jsx'
import About from './pages/About.jsx'
import Charts from './pages/Charts.jsx'
import DisasterAlerts from './pages/DisasterAlerts.jsx'
import ErrorPage from './pages/ErrorPage.jsx'
import Irrigation from './pages/Irrigation.jsx'
import Contact from './pages/Contact.jsx'
import CropSuggestion from './pages/CropSuggestion.jsx'
import Chatbot from './pages/Chatbot.jsx'
import CropsData from './pages/CropsData.jsx'
import Connect from './pages/Connect.jsx'
import Gallery from './pages/Gallery.jsx'
import WaterManagement from './pages/WaterConservation.jsx'
import Disease from './pages/Disease.jsx'
const router = createBrowserRouter([ {path: '/', element: <Homepage
/>,errorElement: <ErrorPage />},
  {path: '/About', element: <About />},
  {path: '/Charts', element: <Charts />},
  {path: '/DisasterAlerts', element: <DisasterAlerts />},
  {path: '/Irrigation', element: <Irrigation />},
  {path: '/contact', element: <Contact/>},
  {path: '/CropSuggestion', element: <CropSuggestion />},
  {path: '/chatbot', element: <Chatbot />},
  {path: '/cropdata', element: <CropsData />},
  {path: '/Expert', element: <Connect/>},
  {path: '/gallery', element: <Gallery />},
  {path: '/WaterManagement', element: <WaterManagement />},
  {path: '/disease-detection', element: <Disease />},
  {path: '*', element: <ErrorPage />}
])
createRoot(document.getElementById('root')).render(
  <StrictMode>
    <RouterProvider router={router}>
      <App />
    </RouterProvider>
  </StrictMode>,
)
```

App.jsx

```
import React from 'react'
const App = () => {
  return (
    <div> </div>
  )
}
export default App
```

Homepage.jsx

```
import React from 'react';
import { useNavigate } from 'react-router-dom';
import { motion } from 'framer-motion';
import { FaChartLine, FaBell, FaTint, FaMicroscope, FaSeedling, FaDatabase,
FaUsers, FaHandHoldingWater, FaRobot } from 'react-icons/fa';
import Navbar from '../components/Navbar'; import WeatherCard from
'../components/WeatherCard';
const FeatureIcon = ({ icon: Icon, label, path, description }) => {
  const navigate = useNavigate();
  return (
    <motion.div
      whileHover={{ scale: 1.05, y: -5 }}
      whileTap={{ scale: 0.95 }}
      onClick={() => navigate(path)}
      className="flex flex-col p-6 rounded-2xl cursor-pointer bg-gradient-to-br
from-green-400/10 via-blue-500/10 to-purple-500/10 backdrop-blur-xl border
border-white/10 shadow-lg hover:shadow-xl hover:shadow-green-500/20
hover:border-white/20 transition-all duration-300">
      <motion.div
        animate={{ y: [0, -5, 0] }}
        transition={{ duration: 2, repeat: Infinity }}
        className="text-5xl mb-4 text-white" >
        <Icon />
      </motion.div>
      <h3 className="text-lg font-semibold text-white mb-2">{label}</h3>
      <p className="text-sm text-white/60">{description}</p>
    </motion.div>
  );
};
const Homepage = () => {
  const navigate = useNavigate();
  const features = [
```

```

    { icon: FaChartLine, label: 'Analytics', path: '/Charts', description: 'View detailed
agricultural analytics and trends' },
    { icon: FaBell, label: 'Alerts', path: '/DisasterAlerts', description: 'Get real-time
disaster and weather alerts' },
    { icon: FaTint, label: 'Irrigation', path: '/Irrigation', description: 'Smart irrigation
control and monitoring' },
    { icon: FaSeedling, label: 'Crop Guide', path: '/CropSuggestion', description:
'Get personalized crop suggestions' },
    { icon: FaMicroscope, label: 'Disease Detection', path: '/disease-detection',
description: 'AI-powered plant disease detection' },
    { icon: FaDatabase, label: 'Crop Data', path: '/cropdata', description: 'Access
comprehensive crop database' },
    { icon: FaUsers, label: 'Connect', path: '/Expert', description: 'Connect with
agriculture experts' },
    { icon: FaHandHoldingWater, label: 'Water Management', path:
'/WaterManagement', description: 'Efficient water conservation techniques' }
  ];
  return (
    <motion.div initial={{ opacity: 0 }} animate={{ opacity: 1 }} className="min-
h-screen pt-20 bg-gradient-to-br from-black via-gray-900 to-black relative">
      <Navbar />
      <div className="container mx-auto px-4">
        <motion.div initial={{ y: 20, opacity: 0 }} animate={{ y: 0, opacity: 1 }}
className="text-center mb-8">
          <h1 className="text-5xl md:text-6xl font-bold mb-4 bg-gradient-to-r from-
green-400 via-blue-500 to-purple-500 bg-clip-text text-transparent">Welcome to
AgriGuard</h1>
          <p className="text-xl text-white/60">Your Complete Smart Farming
Solution</p>
        </motion.div>
        <div className="mb-12"><WeatherCard /></div>
        <motion.div variants={container} initial="hidden" animate="show"
className="grid grid-cols-1 sm:grid-cols-2 lg:grid-cols-3 xl:grid-cols-4 gap-6
max-w-7xl mx-auto pb-24">
          {features.map((feature, index) => (
            <motion.div key={index} variants={item} custom={index}>
              <FeatureIcon {...feature} />
            </motion.div>
          ))}
        </motion.div>
      </div>
      <motion.div
        whileHover={{ scale: 1.1 }}
        whileTap={{ scale: 0.9 }}
        className="fixed bottom-6 right-6 bg-gradient-to-r from-green-400 to-blue-
500 rounded-full p-4 shadow-lg cursor-pointer hover:shadow-xl transition-all
duration-300 z-50"
        onClick={() => navigate('/chatbot')}

```

```

    >
    <FaRobot className="text-white text-2xl" />
    <span className="absolute -top-2 -right-2 bg-red-500 text-white text-xs
rounded-full w-5 h-5 flex items-center justify-center">1</span>
  </motion.div>
</motion.div>
);
};
const container = {hidden: { opacity: 0 }, show: { opacity: 1, transition: {
staggerChildren: 0.1 } }};
const item = {hidden: { y: 20, opacity: 0 }, show: i => ({ y: 0, opacity: 1, transition:
{ delay: i * 0.1 } })};
export default Homepage;

```

About.jsx

```

import React from 'react';
import { motion } from 'framer-motion';
import Navbar from '../components/Navbar';
import { FaLeaf, FaSeedling, FaHandHoldingHeart } from 'react-icons/fa';
const About = () => (
  <motion.div initial={{ opacity: 0 }} animate={{ opacity: 1 }}
  className="min-h-screen pt-20 bg-gradient-to-br from-black via-
gray-900 to-black">
    <Navbar />
    <div className="relative h-[400px] mb-16">
      <div className="absolute inset-0">
        
        <div className="absolute inset-0 bg-gradient-to-b from-black/60
to-black/90" />
      </div>
      <div className="relative container mx-auto px-4 h-full flex items-
center">
        <motion.div initial={{ y: 20, opacity: 0 }} animate={{ y: 0,
opacity: 1 }} className="max-w-2xl">
          <h1 className="text-5xl font-bold mb-6 bg-gradient-to-r from-
green-400 via-blue-500 to-purple-500 bg-clip-text text-
transparent">Revolutionizing Agriculture</h1>
          <p className="text-xl text-white/80 leading-
relaxed">AgroTech is committed to transforming agriculture through

```

innovative technology solutions, empowering farmers to build a sustainable and prosperous future.</p>
 </motion.div>
 </div>
 </div>
 <div className="container mx-auto px-4 py-16">
 <div className="grid md:grid-cols-2 gap-12 items-center">
 <motion.div initial={{ x: -20, opacity: 0 }} animate={{ x: 0, opacity: 1 }} transition={{ delay: 0.2 }}>
 <h2 className="text-3xl font-semibold text-white mb-6">Our Mission</h2>
 <p className="text-white/60 leading-relaxed mb-6">At AgroTech, we believe in the power of technology to transform agriculture. Our mission is to provide farmers with innovative tools and expertise that enable sustainable farming practices and improved yields.</p>
 <div className="flex items-center gap-2 text-green-400"><FaLeaf />Sustainable Agriculture</div>
 </motion.div>
 <motion.div initial={{ x: 20, opacity: 0 }} animate={{ x: 0, opacity: 1 }} transition={{ delay: 0.3 }} className="rounded-2xl overflow-hidden">

 </motion.div>
 </div>
 </div>
 <div className="bg-white/5 backdrop-blur-xl py-16">
 <div className="container mx-auto px-4">
 <motion.h2 initial={{ y: 20, opacity: 0 }} animate={{ y: 0, opacity: 1 }} className="text-3xl font-semibold text-white text-center mb-12">Our Values</motion.h2>
 <div className="grid md:grid-cols-3 gap-8">
 {[
 { icon: FaSeedling, title: 'Innovation', description: 'Constantly evolving and adapting to bring the latest agricultural technologies.' },
 { icon: FaHandHoldingHeart, title: 'Empowerment', description: 'Supporting farmers with knowledge and tools for better decision-making.' },

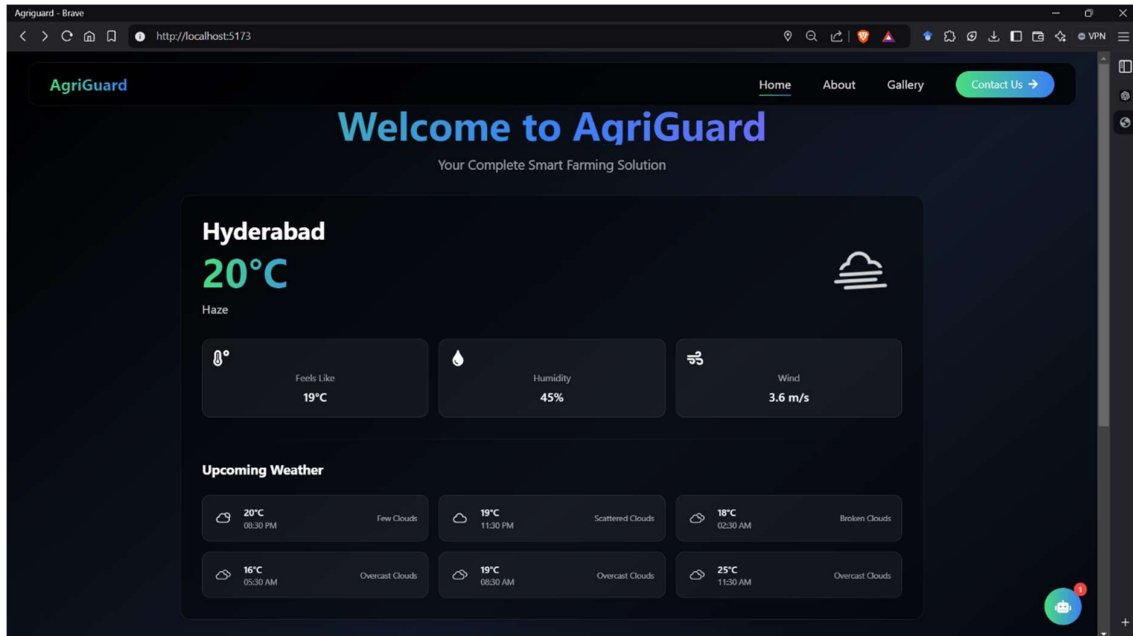
```

    { icon: FaLeaf, title: 'Sustainability', description: 'Promoting
eco-friendly farming practices for a better tomorrow.' }
  ].map((value, index) => (
    <motion.div key={index} initial={{ y: 20, opacity: 0 }}
animate={{ y: 0, opacity: 1 }} transition={{ delay: 0.4 + index * 0.1
}} className="text-center">
      <value.icon className="text-4xl text-green-400 mb-4 mx-
auto" />
      <h3 className="text-xl font-semibold text-white mb-
3">{value.title}</h3>
      <p className="text-white/60">{value.description}</p>
    </motion.div>
  )))
</div>
</div>
</div>
<motion.div initial={{ y: 20, opacity: 0 }} animate={{ y: 0, opacity:
1 }} transition={{ delay: 0.8 }} className="container mx-auto px-4
py-16 text-center">
  <h2 className="text-3xl font-semibold text-white mb-6">Join Us
in Transforming Agriculture</h2>
  <p className="text-white/60 max-w-2xl mx-auto mb-
8">Together, we can build a more sustainable and profitable future for
agriculture. Start your journey with AgroTech today.</p>
  <button className="px-8 py-3 bg-gradient-to-r from-green-400
to-blue-500 rounded-lg text-white font-semibold hover:opacity-90
transition-opacity">Get Started</button>
</motion.div>
</motion.div>
);
export default About;

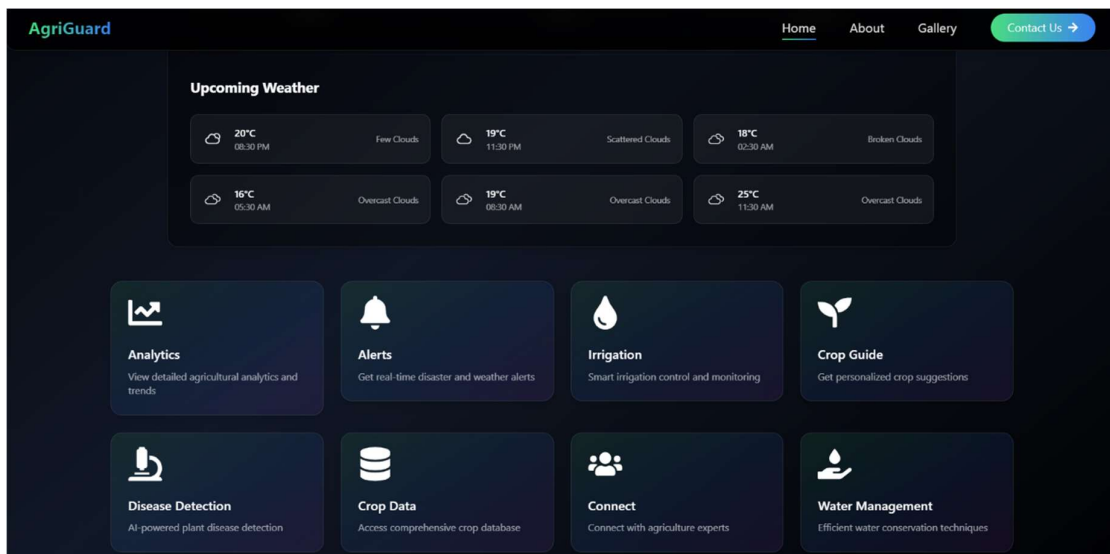
```

4. RESULTS

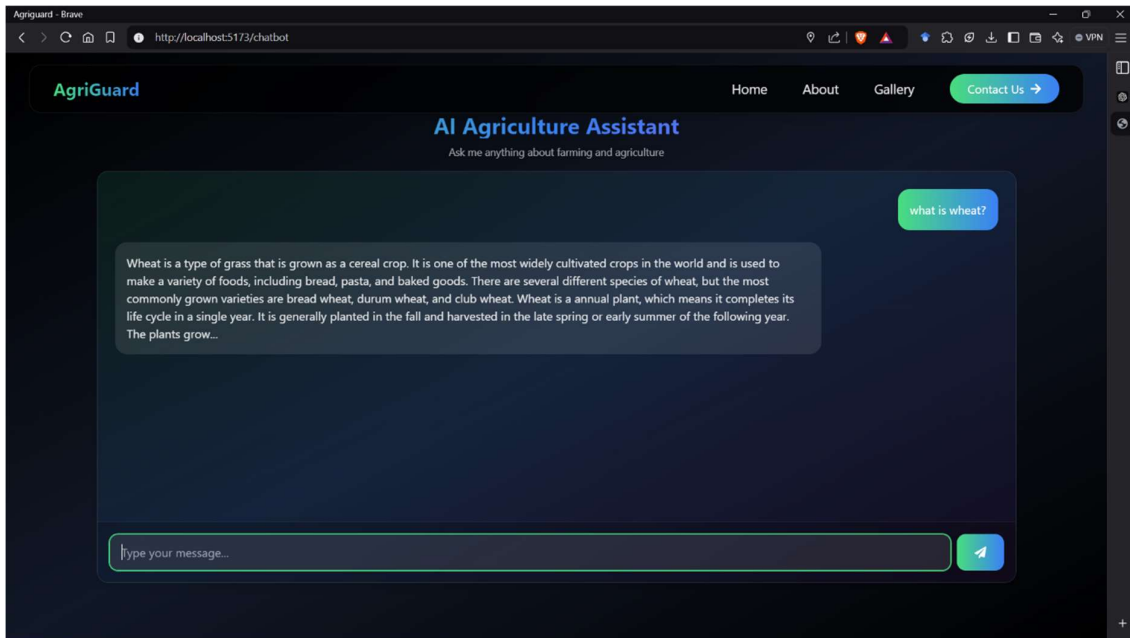
Webpage Start interface



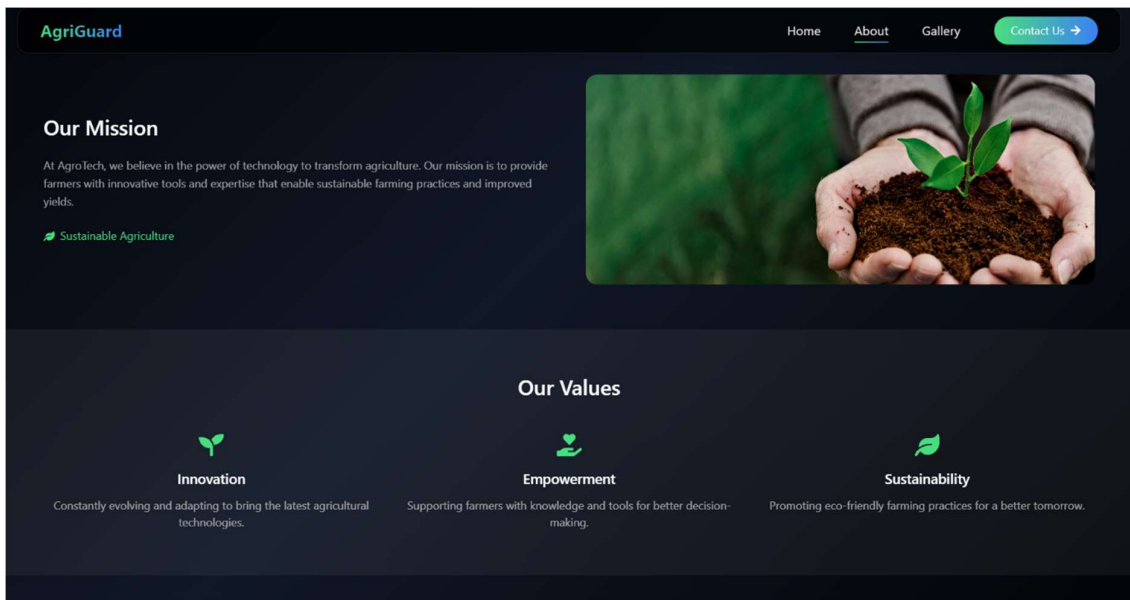
Homepage



ChatBot



About



5. CONCLUSION

The Minimalistic Blogging Platform effectively simplifies blogging by combining voice recognition, AI-powered assistance, and real-time collaboration into a single application. Built with modern web technologies, it addresses the challenges faced by bloggers, making content creation more efficient and accessible.

The platform's modular and scalable design supports future enhancements, such as multilingual support and advanced security features, ensuring adaptability for various use cases. This project highlights the potential of integrating cutting-edge technology into blogging, paving the way for innovation in digital content creation.

6. REFERENCES

- NASA POWER API Documentation
NASA. (n.d.). *POWER API - Prediction Of Worldwide Energy Resources*. Retrieved from <https://power.larc.nasa.gov/>
- Open Meteo API Documentation
Open-Meteo. (n.d.). *Open-Meteo Weather API*. Retrieved from <https://open-meteo.com/>
- OpenWeather API Documentation
OpenWeather. (n.d.). *Weather API Documentation*. Retrieved from <https://openweathermap.org/api>
- Hugging Face Transformers Documentation
Hugging Face. (n.d.). *Transformers Library Documentation*. Retrieved from <https://huggingface.co/docs/transformers/>
- React Documentation
Meta. (n.d.). *React – A JavaScript Library for Building User Interfaces*. Retrieved from <https://react.dev/>
- Tailwind CSS Documentation
Tailwind Labs. (n.d.). *Tailwind CSS Documentation*. Retrieved from <https://tailwindcss.com/docs>
- Recharts Documentation
Recharts. (n.d.). *Recharts Documentation*. Retrieved from <https://recharts.org/en-US>
- Framer Motion Documentation
Framer. (n.d.). *Framer Motion – React Animation Library*. Retrieved from <https://www.framer.com/motion/>
- Firebase Realtime Database Documentation
Firebase. (n.d.). *Firebase Realtime Database*. Retrieved from <https://firebase.google.com/docs/database>
- Node.js Documentation
OpenJS Foundation. (n.d.). *Node.js Documentation*. Retrieved from <https://nodejs.org/en/docs/>