LLMediCare: AI-Powered Healthcare Assistant

Team members name and email id (1st should be TL)	Aditya Gupta - aditya22031@iiitd.ac.in Sahil Gupta - sahil22430@iiitd.ac.in Debjit Banerji - debjit22146@iiitd.ac.in Kartik Prasad - kartik22240@iiitd.ac.in
Sponsor (Name, title and affiliation, email)	Name - Ravi Anand Title and affiliation - Assistant Professor (CSE), IIITD Email - ravi.anand@iiitd.ac.in Name -Pragya Jha Title and affiliation-Assistant Professor(Data Science and Computer Applications) , Manipal Institute of Technology , Manipal, Udupi, Karnataka Email - pragya.jha@manipal.edu
A bit about the sponsor / its org	The project is being undertaken under the guidance of Professor Ravi Anand, Assistant Professor in the Department of Computer Science and Engineering at the Indraprastha Institute of Information Technology, Delhi (IIITD), and Professor Pragya Jha, Assistant Professor in the Department of Data Science and Computer Applications at the Manipal Institute of Technology, Manipal, Karnataka. Professor Pragya Jha will also be responsible for testing the website developed during the project.
Users of the application	LLMediCare serves as a comprehensive healthcare platform designed to empower both patients and healthcare professionals with advanced tools and seamless interactions. For everyday users, it provides features like personalized health recommendations, AI-powered symptom checking, and effortless appointment scheduling, all accessible via a user-friendly interface. Patients can securely manage their medical records, receive medication reminders, and access educational resources tailored to their conditions. By leveraging cutting-edge AI and NLP capabilities, the platform ensures users receive timely and accurate healthcare assistance, improving overall health management. Its focus on accessibility also bridges the gap for underserved populations through virtual consultations and real-time medical advice. For healthcare professionals, LLMediCare streamlines patient management and enhances decision-making through AI-powered clinical insights and analytics. Doctors can efficiently manage appointments, view patient records, and securely share prescriptions or results with their patients. Natural language query processing enables quick data retrieval, while dashboards provide valuable insights into patient trends and treatment outcomes. The platform's integration with Microsoft Azure ensures secure, scalable, and resilient operations, while automated workflows and real-time communication tools reduce administrative burdens. By combining advanced technology with practical usability, LLMediCare transforms the healthcare experience for both providers and patients alike.
Brief description of the project and key requirements	**Brief Description** **LLMediCare: AI-Powered Healthcare Assistant** is an advanced healthcare management platform designed to provide seamless patient data management, appointment scheduling, and AI-powered medical assistance. The system will use RESTful APIs to deliver backend services for web and mobile clients, integrating cutting-edge AI and NLP capabilities to assist users with personalized healthcare recommendations. The platform will focus on data security,

	scalability, and ease of deployment, leveraging Microsoft Azure cloud services for reliable and resilient operations. **Key Requirements** 1.Develop robust RESTful APIs using Django REST Framework (DRF) with MySQL as the primary database for managing healthcare data. Integrate AI/ML models (TensorFlow or PyTorch) and NLP tools (spaCy or Hugging Face Transformers) to provide intelligent healthcare recommendations and natural language query processing. 2.Build a dynamic and user-friendly interface using modern JavaScript frameworks like React.js, Angular, or Vue.js. Ensure real-time interaction with the backend through REST APIs, providing seamless experiences for patients and administrators. 3.Containerize the application using Docker and deploy it on Microsoft Azure, leveraging services like Azure App Service or AKS for scalability and reliability. Use Azure Database for MySQL for secure and high-performance database management, with CI/CD pipelines for automated testing and deployments. 4.Implement token-based authentication (or JWT) for secure user access and role-based permissions. Optimize performance with caching (Redis or Memcached) and MySQL query tuning. Use Azure Monitor and Application Insights for real-time performance tracking and system logging.
Tech stack	**Backend** The backend of LLMediCare will be developed using Python and Django, with Django REST Framework (DRF) providing a robust foundation for RESTful API development. Advanced AI and ML capabilities will be integrated using TensorFlow or PyTorch, while NLP tools like spaCy or Hugging Face Transformers will handle intelligent natural language processing tasks. **MySQL** will serve as the production database for reliable and scalable data management, ensuring compatibility with the application's requirements. Secure token-based authentication (or JWT) will protect user interactions and access. **DevOps & Deployment (Microsoft Azure)** Deployment will be carried out on Microsoft Azure to leverage its enterprise-grade cloud services for scalability and resilience. The application will be containerized using Docker and Docker Compose, with Gunicorn as the WSGI server and Nginx as a reverse proxy to enhance security and performance. Continuous integration and deployment (CI/CD) pipelines will be set up using Azure DevOps or GitHub Actions to automate testing, builds, and deployments. Azure Container Registry will manage container images, while hosting will be handled via Azure App Service or Azure Kubernetes Service (AKS). **MySQL** will be deployed as a managed service using Azure Database for MySQL for seamless integration with the application. **Frontend** The frontend of the application will be built using modern JavaScript frameworks such as React.js, Angular, or Vue.js to deliver a responsive and intuitive user interface. Communication with the backend APIs will be facilitated using Axios or the Fetch API, ensuring real-time data updates and seamless user interactions. This will allow both end users and administrators to interact with the platform effortlessly, improving accessibility and usability. **Additional Tools and Services** Version control will be managed using Git, ensuring collaboration through platforms like GitHub or Azure Repos. Virtual environment tools such as virtualenv or pipenv will be used to ensure consistent dependency management across environments. Comprehensive testing will be conducted using Django's

	built-in testing tools along with pytest and pytest-django to ensure high-quality code. Asynchronous tasks will be handled by Celery, with RabbitMQ or Azure Service Bus as the message broker, while Redis or Memcached will be employed for caching to enhance performance and responsiveness. **MySQL** database performance will be optimized using connection pooling and query tuning for fast and efficient data access.