Requirement Gathering and Analysis Phase
Technology Stack (Architecture & Stack)

Date	22 June 2024
Team ID	PNT2022TMID1720113552
Project Name	Project - Video Conferencing Application
Maximum Marks	3 Marks

Technical Architecture:

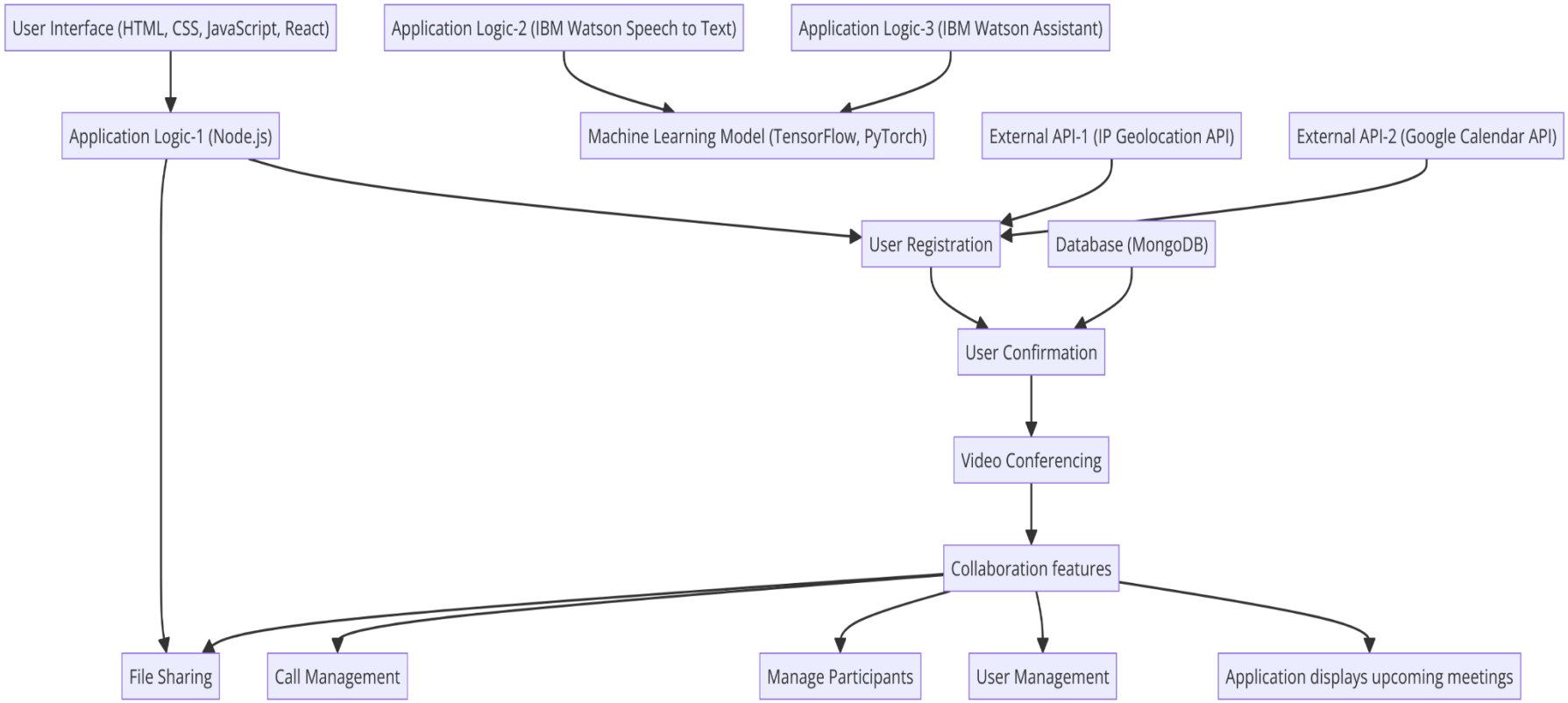

Table-1 : Components & Technologies:

S.no:	Component	Description	Technology
1	User Interface	Web-based interface for initiating calls, managing participants, and video meetings.	HTML, CSS, JavaScript (React)
2	Application Logic-1	Handles call management, user interactions, and core functionalities.	Node.js (with Express.js framework)
3	Application Logic-2	Processes speech-to-text conversion for features like voice commands or captions.	IBM Watson Speech to Text service
4	Application Logic-3	Provides chat functionality for user assistance and basic interactions.	IBM Watson Assistant
5	Database	Stores user information, call details, configurations, and potentially recording data.	MongoDB (NoSQL database)
6	Cloud Database (Optional)	Stores data in a managed cloud database service for scalability and reliability.	MongoDB Atlas (Cloud-based MongoDB)
7	File Storage	Stores recordings, user files, or other application-specific data.	IBM Block Storage (Cloud), Other Cloud Storage Service, or Local Filesystem
8	External API-1	Integrates time zone API retrieve the time zones of participants, simplifying scheduling meetings.	IP Geolocation API

9	External API-2	Integrates Calendars to create add meetings	Google Calender API
10	Machine Learning Model	Integrates an object recognition model for background blurring or other features.	TensorFlow, PyTorch
11	Infrastructure (Server / Cloud)	Local Server Configuration: Deployment on a local server (for testing or small-scale use). Cloud Server Configuration: Deployment on a cloud platform for scalability and wider audience.	Local Operating System. Cloud Foundry, Kubernetes (Orchestration platforms)

Table-2: Application Characteristics:

S.No.:	Characteristics	Description	Technology
1	Open-Source Frameworks	Utilize open-source frameworks for UI development and backend logic (MERN Stack).	React, Node.js (Express.js), MongoDB
2	Security Implementations	Data encryption at rest and in transit, Secure user authentication and authorization and IAM (Identity and Access Management) controls	AES- 256 and JWT

3	Scalable Architecture	Utilize a microservices architecture with a horizontally scalable database.	Node.js with microservice frameworks (e.g., NestJS)
4	Availability	- Load balancers to distribute traffic across multiple Node.js instances.	Technologies like Nginx, HAProxy
5	Performance	- Caching mechanisms for frequently accessed data (e.g., in-memory caching with Redis) - Content Delivery Networks (CDNs) for efficient content delivery.	- Cloud providers like AWS, Azure offer CDN services