



Contact  
[recrutement@quicktext.im](mailto:recrutement@quicktext.im)

# PFE BOOK 2025



# Mobile Voice Assistant

**Duration**  
4 - 6 months

## Objective

Develop a mobile application that allows hotel guests to control various hotel services and get information using voice commands.

## Analysis:

Study client requirements, the needs of hotel guests, and analyze existing voice assistants like Google Assistant, Alexa, and Siri, particularly their use in hospitality. Investigate how these technologies can improve guest experiences and streamline hotel management processes.

- Hotel Services: Room service, housekeeping requests, concierge services, etc.
- Guest Experience: How voice can enhance comfort, accessibility, and efficiency for guests.

## Technologies:

- **Mobile Development:** Expo (React Native), or native mobile development for iOS/Android
- **Voice Command Processing:** Voice processing API integrations (UX similar to ChatGPT Voice Feature, Google Assistant, Alexa, Siri ...)
- **Speech-to-Text (STT):** Convert guest voice commands to text.
- **Text-to-Speech (TTS):** Provide audible responses to guests.
- **Conversational AI:** Use AI models for understanding and responding to guest queries (e.g., GPT-4, RAG) or / and use Quicktext AI.
- **Real-Time Communication:** WebSockets or alternative for live communication between the app and hotel systems



- **Backend:** RESTful services, Cloud hosting (e.g., AWS, Azure)
- **AI & Machine Learning:** Personalized responses based on guest history and preferences
- **Security:** Authentication (biometric, voice recognition) and data privacy

**Duration**  
4 - 6 months

#### **Expected Deliverables:**

- A fully functional mobile application with voice-command features.
- Clean and ergonomic UX & UI.
- Documentation detailing the system architecture, API integrations, and user instructions.
- A demo video or live demonstration of the app.

# Business Intelligence Solution

**Duration**  
4 - 6 months

## Objective

Optimize the existing Business Intelligence (BI) solution and develop a Data Warehouse to improve data processing, reporting, and decision-making capabilities.

## Analysis:

Study the current BI solution, evaluate performance bottlenecks, and identify key areas for improvement. Analyze data flow, storage mechanisms, and reporting processes to determine inefficiencies. Investigate the best approach for developing a robust Data Warehouse that integrates data from various sources and enhances analytical capabilities.

- BI Solution Analysis: Assess data processing speed, report generation times, and data integration issues.
- Data Warehouse Design: Focus on structuring the data warehouse for optimal storage, easy retrieval, and scalability.
- Data Reporting & Insights: Improve reporting systems to provide more actionable insights for better business decision-making.
- Scalability: Ensure the solution can scale as data volume grows over time.

## Technologies:

- **Database:** PostgreSQL (for efficient storage and query processing)
- **Backend:** NestJS (for building a scalable and maintainable API)
- **API Communication:** GraphQL (for optimized data retrieval)



- **ETL Process:** Tools like Apache NiFi or custom ETL scripts to manage data extraction, transformation, and loading.
- **Data Warehouse:** Implement a star schema or snowflake schema for structuring the Data Warehouse.

**Duration**  
4 - 6 months

#### **Expected Deliverables:**

- An optimized BI solution that delivers faster, more accurate reports.
- A fully functional Data Warehouse that integrates with the existing BI system.
- Documentation detailing the architecture, ETL processes, and query optimization techniques.
- A live demonstration or demo video showcasing the improved BI solution and Data Warehouse.

# Dynamic Content Chat Service

**Duration**  
4 - 6 months

## Objective

Design and develop a chat service that supports dynamic content, enabling real-time communication and rich media exchange between users.

## Analysis:

Investigate user requirements and assess existing chat solutions to determine best practices and features that should be implemented. Analyze the need for dynamic content support (e.g., images, videos, file attachments, emojis) within chat conversations. Evaluate scalability, performance, and security concerns.

- User Requirements: Identify essential features for users, such as real-time messaging, multimedia support, and notifications.
- Existing Chat Solutions: Study successful platforms (e.g., WhatsApp, Slack, Discord) to understand their architecture, features, and technologies.
- Dynamic Content: Determine how to handle and display images, videos, and other types of media within chat messages.
- Scalability & Performance: Ensure the system can handle a large number of simultaneous users and data traffic efficiently.
- Security: Implement end-to-end encryption and secure file sharing.

## Technologies:

- **Real-Time Communication:** WebSocket (for real-time message delivery and notifications)
- **Backend Framework:** NestJS (for building a scalable, modular API)
- **Frontend:** React (for building a responsive and interactive user interface)



- **API Communication:** GraphQL (for efficient and flexible data retrieval)
- **Database:** MongoDB (for storing chat history, user data, and dynamic content)
- **Search Engine:** ElasticSearch (for fast and efficient searching of messages, files, and user activity)

**Duration**  
4 - 6 months

#### **Expected Deliverables:**

- A fully functional chat service with real-time messaging, dynamic content support (images, videos, files), and search functionality.
- A user-friendly and responsive React-based frontend.
- A scalable backend built using NestJS, WebSocket, GraphQL, and MongoDB.
- Documentation detailing the system architecture, implementation, and API usage.
- A demo video or live demonstration of the chat service in action.

# Email Management Assistant

**Duration**  
4 - 6 months

## Objective

Develop an Outlook add-on for improved email management, enhancing productivity through automated organization, categorization, and intelligent prioritization of emails.

## Analysis:

Analyze current email management workflows within Outlook to identify pain points and areas for improvement. Understand how users typically organize, prioritize, and respond to emails. Investigate common issues like email overload, missed important messages, and inefficient categorization. Explore how automation and intelligent assistance can improve user efficiency.

- Email Management Workflows: Study how users currently manage their inbox, including categorization, prioritization, and search.
- Pain Points: Identify challenges in email overload, missed important messages, and time-consuming manual processes.
- Automation & AI Integration: Explore opportunities for automating tasks like sorting, flagging, and categorizing emails using AI models.
- Productivity Features: Identify features that can save time, such as automatic labeling, email summarization, or intelligent filtering.

## Technologies:

- **Outlook Add-ons:** Utilize Outlook's Add-in framework to integrate the add-on directly into the Outlook interface.
- **Backend Development:** NestJS (for building scalable backend services to process and analyze emails)
- **Email Automation:** JavaScript (for implementing frontend and integration with Outlook)
- **AI Integration:** RAG (Retrieve and Generate models for email summarization, priority analysis, and categorization)



### Expected Deliverables:

- A fully functional Outlook add-on that improves email management by automating tasks like categorization, prioritization, and summarization.
- A user-friendly interface integrated into Outlook for easy access and interaction with the add-on.
- A scalable backend (using NestJS) to support email processing and AI-driven features.
- Documentation on installation, setup, and usage of the add-on.
- A demo video or live demonstration of the add-on in action, showcasing its features and impact on email management.

### Duration

4 - 6 months

# HR Management Web Application

**Duration**  
4 - 6 months

## Objective

Design a web application to streamline and automate HR processes, enhancing the management of employee data, recruitment, onboarding, performance evaluations, and other HR functions.

## Analysis:

Study the needs of HR departments and evaluate existing HR management tools to identify gaps and opportunities for improvement. Understand the specific HR processes that need to be automated, such as employee record management, payroll, performance tracking, and recruitment workflows. Analyze the potential for using AI-driven features, such as candidate screening or performance prediction, to improve efficiency and decision-making.

- **HR Needs:** Assess core HR functions like recruitment, onboarding, performance management, payroll, employee engagement, and data privacy.
- **Existing HR Tools:** Review current HR management tools (e.g., Workday, BambooHR, SAP SuccessFactors) to understand their strengths and weaknesses.
- **Process Automation:** Identify opportunities to automate and optimize key HR processes to reduce manual effort and errors.
- **AI in HR:** Investigate how RAG (Retrieve and Generate) models or other AI technologies can be used for tasks like candidate screening, performance analysis, or employee engagement insights.

## Technologies:

- **Backend Framework:** NestJS (for building a scalable, maintainable backend)
- **Frontend Development:** React (for building an intuitive, responsive user interface)



- **Database:** PostgreSQL (for managing employee records, performance data, and other HR-related information)
- **AI Integration:** RAG (Retrieve and Generate models for automating HR processes like candidate screening or performance evaluation)
- **HR Tech:** Use HR-specific tools and libraries for payroll, compliance tracking, and other HR functionalities.

**Duration**  
4 - 6 months

#### **Expected Deliverables:**

- A fully functional web application that streamlines HR processes such as recruitment, performance management, and employee record handling.
- A responsive React-based frontend that provides an intuitive user experience for HR professionals and employees.
- A backend built with NestJS that handles HR workflows, integrates AI features, and stores employee data in PostgreSQL.
- Integration of AI technologies (RAG) for automating and enhancing HR decision-making, such as recruitment and performance reviews.
- Documentation detailing the system architecture, API integrations, and how to use the web application.
- A demo video or live demonstration of the web application showcasing its HR process automation and management features.

# Customer Support Mobile Chat App

**Duration**  
4 - 6 months

## Objective

Create an AI model for voice-guided room reservation, allowing users to book rooms through natural language voice commands, providing an intuitive and hands-free booking experience.

## Analysis:

Investigate the needs of users who require voice-guided room booking systems and analyze existing solutions in the hospitality industry. Understand how users interact with current reservation systems and identify pain points. Evaluate how voice assistants (like Alexa, Google Assistant, and Siri) are utilized for similar purposes and explore opportunities to improve the user experience through advanced AI models.

- User Needs: Identify user requirements for room reservation systems, such as ease of use, speed, and accessibility.
- Voice Interaction: Explore how voice can be used effectively to make booking decisions, inquire about room availability, and provide relevant hotel information.
- Existing Solutions: Analyze voice-guided booking solutions available in the market, understanding their limitations and strengths.
- AI Integration: Determine how conversational AI, GPT models, and voice-to-text (STT) systems can be used to enhance user interactions and streamline the booking process.

## Technologies:

- **Mobile Development:** Expo (for building cross-platform mobile applications with React Native)
- **Voice Command Processing:** TTS (Text-to-Speech), STT (Speech-to-Text) for converting voice commands to text and responding via voice.



- **Backend Integration:** API (to integrate with hotel booking systems and retrieve room availability, prices, etc.)
- **AI & Machine Learning:** GPT (for understanding and generating conversational responses) and RAG (Retrieve and Generate) models for booking logic and personalized responses.
- **Real-Time Communication:** Socket (for real-time communication between the mobile app and backend systems)
- **Programming Language:** Python (for backend logic, machine learning, and AI model integration)

**Duration**  
4 - 6 months

#### **Expected Deliverables:**

- A fully functional mobile application (developed using Expo) that allows users to book rooms via voice commands.
- Integration of AI models (GPT for natural language understanding and RAG for generating booking information).
- Real-time communication between the app and backend systems using Socket for live data exchange.
- A backend system developed in Python to handle voice commands, AI processing, and room reservation functionalities.
- Documentation detailing the system architecture, API integrations, and instructions for using the voice-guided room reservation system.
- A demo video or live demonstration showcasing the complete booking process from voice command to room reservation.

# AI-Powered Voicie-Guided Room Booking System

Duration  
4 - 6 months

## Objective

Create an AI model for a voice-guided room reservation system that allows users to make reservations seamlessly using natural language voice commands, offering a hands-free and intuitive booking experience.

## Analysis:

Investigate user needs for voice-guided booking systems in the context of hotel room reservations. Analyze existing solutions such as voice assistants (Google Assistant, Alexa, Siri) and how they are used in the hospitality industry. Understand the challenges of current booking systems and identify areas where voice interaction can improve user experience. Explore AI models for voice recognition and conversational management, with an emphasis on customer engagement and service automation.

- User Needs: Determine the features users expect from a voice-guided room reservation system (e.g., availability checks, price quotes, booking confirmation).
- Voice Interaction: Study how voice commands can simplify the booking process, reduce friction, and enhance accessibility for users with disabilities.
- Existing Solutions: Analyze existing voice-guided solutions in hospitality and other industries to understand their strengths and limitations.
- AI Integration: Investigate AI-driven functionalities such as conversational AI for booking management, automatic room availability checks, and intelligent query handling.

## Technologies:

- **Frontend Development:** Expo (for cross-platform mobile application development using React Native).
- **Voice Command Processing:** TTS (Text-to-Speech) and STT (Speech-to-Text) for converting voice to text and generating natural spoken responses.
- **Backend Development:** Python (for the backend logic, integrating APIs, and AI models).



- **AI & Machine Learning:** GPT (Generative Pretrained Transformer) for conversational responses, RAG (Retrieve and Generate) models for intelligent room booking logic.
- **Real-Time Communication:** Socket (for real-time communication between the mobile app and backend services).
- **API Integration:** Use APIs for retrieving room availability, prices, and making reservations.

**Duration**  
4 - 6 months

#### **Expected Deliverables:**

- A fully functional voice-guided room reservation mobile application with voice recognition and conversational booking.
- Integration of AI models like GPT for conversation handling and RAG for intelligent responses and retrieval of room availability.
- Backend built with Python that manages room availability checks, bookings, and customer queries.
- Real-time communication between the mobile app and backend using socket technology.
- Documentation on the system architecture, AI model usage, and how to use the app.
- A demo video or live demonstration of the voice-guided room reservation system, showcasing its functionality.

# AI-Powered RAG Chatbot for Productivity Optimization and Role-Based Data Access in Microsoft Teams

Duration  
4 - 6 months

## Objective

Develop an AI-powered chatbot for Microsoft Teams using the Retrieval-Augmented Generation (RAG) model to optimize productivity and facilitate data access. The chatbot will respect role-based permissions for each user, ensuring that users only have access to data relevant to their roles, while providing a more intuitive and efficient conversational experience.

## Analysis:

- User Needs: Identify the needs of team members and managers within Microsoft Teams, focusing on how they interact with data, resources, and workflows.
- Data Access: Understand how data access is currently managed within Teams and what the key limitations are with respect to user roles and permissions.
- Current Solutions: Evaluate existing chatbots and conversational agents in Microsoft Teams to determine their effectiveness in handling data retrieval and user interaction.
- Role-Based Permissions: Ensure the system supports dynamic, role-based access control (RBAC) for security, so users only have access to data relevant to their assigned roles within the organization.
- RAG Model: Investigate how RAG (Retrieve and Generate) can be applied to provide accurate, context-specific responses to user queries based on organizational data.



# QUICKTEXT

## Technologies:

- **AI Model:** RAG (Retrieve and Generate) for combining data retrieval and natural language generation to provide relevant, contextual, and personalized responses.
- **Microsoft Teams Integration:** Use Microsoft Graph API to interact with Teams data and ensure seamless integration.
- **Backend Development:** Python or Node.js for backend logic, including querying data repositories and integrating the RAG model.
- **Permissions Management:** Role-Based Access Control (RBAC) to restrict data access based on the user's role within the organization.
- **Conversational AI:** Use conversational AI frameworks (e.g., Microsoft Bot Framework, OpenAI GPT) for processing and generating natural language responses.
- **Frontend:** Microsoft Teams Bot Framework for embedding the chatbot directly into Teams.

## Duration

4 - 6 months

## Expected Deliverables:

- A fully functional AI-powered chatbot integrated with Microsoft Teams that allows users to interact with organizational data through natural language.
- A backend system that retrieves and processes data, respects role-based permissions, and generates accurate responses using the RAG model.
- Documentation detailing the system architecture, user roles, AI model design, and integration with Microsoft Teams.
- A demo or live demonstration of the chatbot, showcasing its capabilities and role-based access features.



To apply, please send your application to

**[recrutement@quicktext.im](mailto:recrutement@quicktext.im)**

specifying the project you have chosen in the subject line.