Content

General Instructions

Dear Candidates,

We're excited about your interest in joining Nebula9.ai for your internship! Our company is at the forefront of applying generative AI and advanced technologies to solve real-world problems, and we're looking for talented individuals like you to contribute to our mission.

As part of the selection process, you have been assigned a project that showcases your skills, creativity, and ability to apply cutting-edge technologies. The project is designed to be completed within 4 days of focused work, but you will have until Friday, October 3rd, to submit your final work.

We have assigned a specific project to each candidate that best aligns with your chosen workstream, interests, and expertise. The complete list of assignments has been shared with your institution's Student Placement Coordinator (SPOC). Please reach out to your coordinator to learn which project has been assigned to you. The project descriptions for all potential assignments are included in this document for your reference.

If our project is shortlisted we will have the final round through a Video Conference, where we discuss the projects that you have done along with your approach/Technique.

We have assigned a project to each student and shared the list with your Institutions SPOC. Please reach out to your coordinator to know about the project that has been assigned to you.

Important Announcement

By design these are open-ended assignments wherein there will be no guidance and the candidates are expected to leverage public data or create data (if needed) to complete this project.

And in case of lack of clarity please take assumptions to solve the roadblock yourself. Candidates who have taken assumptions to solve a problem will have an edge over others during the final Interview Process.

The idea of this approach is to check the problem solving and decision making ability of the candidate. Also we want to evaluate how independently a candidate can execute projects since this represents the DNA of ownership and result orientation. Feel free to utilize any internet resources available to fulfill these assignments.

How you communicate during the final Interview (Presentations Skills and Project Representation) is also important.

Submission Procedure/Details:

Kindly submit your project using the Google Form shared with you by **Friday, Oct 3rd, by 10:00 PM**.

Include a link to your GitHub repository containing all code, documentation, and any deployment instructions or in case of non technical workload ensure that you attach the final output in the email. Ensure your repository is public or share a private link with access for our review team.

Full Stack Developer Internship Assignment - Nebula9.ai

Dear Candidates,

Thank you for your interest in the Full Stack Developer internship position at Nebula9.ai. We are excited about the possibility of having you contribute to our innovative projects that seamlessly integrate generative AI technologies with robust full-stack development practices.

As part of our selection process, we require all candidates to complete a practical assignment. These projects are designed to assess your skills in full-stack development, focusing on API development, authentication and authorization mechanisms, role-based access control (RBAC), cloud deployment, notification management, and the integration of generative AI functionalities. Each assignment is scoped to be feasibly completed within days of work, and you will have 3rd October from the receipt of this email to submit your work.

List of Assignments:

AI-Powered Project Assignment List

1. Multimodal Code Debug Assistant

Objective: Create a developer tool that accepts code, logs, screenshots, and error traces to suggest fixes using LLMs and agentic workflows. AI agents extract errors, retrieve documentation, and provide corrected code with rationales. Multimodal support allows diagram and UI bug analysis.

Key Requirements:

Frontend:

- React interface for code editing, log viewing, and AI suggestions
- Screenshot annotation tools for UI bugs
- Real-time collaboration features

Backend:

- Node.js or FastAPI with RESTful APIs
- PostgreSQL for code history and error patterns
- File upload handling for screenshots and logs

Authentication:

- JWT tokens for session management
- OAuth integration (Google/GitHub) for easy login
- Basic user profile management

AI Integration:

- GPT-4, Claude 3.5, or open-source LLMs for code analysis
- Computer vision models for screenshot analysis
- Multi-agent workflow for error classification and solution generation

Agents (Good to have, though not mandatory):

- AutoGen or LangChain agents for multi-step debugging
- Error extraction agent
- Documentation retrieval agent
- Code correction agent with rationale

Real-Time (Good to have, though not mandatory):

- WebSockets for collaborative debugging sessions
- Live code sharing and synchronized cursors

Deployment (Good to have, though not mandatory):

- Dockerized application
- Cloud-hosted (Heroku, Vercel, or similar student-friendly platforms)
- Basic monitoring and logging

Success Metrics (Good to have, though not mandatory):

- Code fix accuracy rate (>80%)
- Time to resolution improvement
- User satisfaction scores

2. AI Newsroom Collaboration Tool

Objective: Build a collaborative AI content generation platform for journalists. Users input story briefs or notes, and AI drafts articles with images and citations. Multi-agent workflows handle research, writing, editing, and fact-checking.

Key Requirements:

Frontend:

- React interface for story briefs, article editing, and collaboration
- Rich text editor with citation management
- Image integration and caption generation
- Editorial workflow management

Backend:

- Node.js or FastAPI with content management system
- PostgreSQL for articles, sources, and user management
- Integration with news APIs and fact-checking services
- Version control for article drafts

Authentication:

- JWT tokens for session management
- OAuth integration (Google/GitHub) for easy login
- Basic user role management (writer, editor)

AI Integration:

- GPT-4 or Claude 3.5 for article generation
- Specialized models for fact-checking and citation
- Image generation and analysis capabilities
- Source credibility assessment

Agents (Good to have, though not mandatory):

- Research agent for source gathering
- Writing agent for article drafting
- Fact-checking agent for verification
- Editorial agent for style and tone consistency

Real-Time (Good to have, though not mandatory):

- SSE for collaborative editing

- Live comments and suggestions
- Real-time fact-checking alerts

Deployment (Good to have, though not mandatory):

- Dockerized application
- Cloud-hosted (Netlify, Vercel, or Heroku)
- Basic content delivery and scaling

Success Metrics (Good to have, though not mandatory):

- Article quality scores
- Fact-checking accuracy
- Time to publication improvement

3. Enterprise Knowledge Assistant

Objective: Create an AI-powered internal assistant that uses an enterprise knowledge base to answer complex, policy-aware employee questions with contextual understanding of company procedures and regulations.

Key Requirements:

Frontend:

- React interface for knowledge queries and search
- Conversational chat interface
- Knowledge base navigation and exploration
- Admin panel for content management

Backend:

- Node.js or FastAPI with enterprise integrations
- PostgreSQL for user management and query history
- Vector database for knowledge embeddings
- Integration with existing enterprise systems (SharePoint, Confluence)

Authentication:

- JWT tokens for session management
- OAuth integration (Google/Microsoft) for easy login
- Basic department-based access controls

AI Integration:

- GPT-4 or Claude 3.5 for natural language understanding
- RAG (Retrieval-Augmented Generation) for knowledge retrieval
- Policy-aware response generation

Agents (Good to have, though not mandatory):

- Knowledge retrieval agent
- Policy interpretation agent
- Answer synthesis agent
- Escalation agent for complex queries

Real-Time (Good to have, though not mandatory):

- WebSockets for instant responses
- Live knowledge updates

Deployment (Good to have, though not mandatory):

- Dockerized application
- Cloud-hosted (student-friendly platforms)
- Basic security measures and data protection

Success Metrics (Good to have, though not mandatory):

- Query resolution rate

4. Medical Imaging and Report Assistant

Objective: Develop a clinical support tool that processes radiology images and patient data to generate draft reports. AI assists with segmentation, annotation, and diagnosis suggestions based on a RAG system of medical literature.

Key Requirements:

Frontend:

- React interface for image viewing and annotation
- Report generation and editing interface
- Patient data integration dashboard

Backend:

- FastAPI with medical imaging pipeline
- PostgreSQL for patient data and reports
- Medical literature database with RAG system

Authentication:

- JWT tokens for session management
- OAuth integration (Google/GitHub) for easy login
- Basic role-based access (student, instructor, admin)
- Simple user profile management

AI Integration:

- Computer vision models for image analysis
- GPT-4V for multimodal medical reasoning
- Specialized medical LLMs for report generation
- Segmentation and annotation models

Agents (Good to have, though not mandatory):

- Image analysis agent for findings detection
- Literature search agent for relevant studies
- Report generation agent with medical terminology
- Quality assurance agent for accuracy checking

Real-Time (Good to have, though not mandatory):

- WebSockets for collaborative case review
- Live image streaming for consultations
- Real-time notification system

Deployment (Good to have, though not mandatory):

- Dockerized application
- Cloud-hosted (student-friendly platforms)
- Basic data encryption and security measures
- Simple backup system

Compliance & Security (Good to have, though not mandatory):

- Basic data protection practices
- Simulated medical data only
- Educational use disclaimer

- Basic audit logging

Success Metrics (Good to have, though not mandatory):

- Diagnostic accuracy improvement
- Report generation time reduction
- Patient outcome correlation

5. Real-Time Financial Insights Dashboard

Objective: Build a real-time financial monitoring system that streams market data and generates insights with GenAI. Users can query markets and get grounded, live summaries and predictions with risk assessment.

Key Requirements:

Frontend:

- React interface with real-time charts and visualizations
- Natural language query interface
- Portfolio tracking and analysis
- Alert and notification system

Backend:

- Node.js or FastAPI with financial data pipeline
- PostgreSQL for user portfolios and historical data
- Time-series database for market data
- Integration with financial data providers

Authentication:

- JWT tokens for session management
- OAuth integration (Google/GitHub) for easy login
- Basic user profile and portfolio management

AI Integration:

- GPT-4 or Claude 3.5 for financial analysis
- Time-series forecasting models
- Sentiment analysis for market news

Agents (Good to have, though not mandatory):

- Market data ingestion agent
- Analysis and insight generation agent
- Risk assessment agent
- Alert and notification agent

Real-Time (Good to have, though not mandatory):

- WebSockets for live market data streaming
- Real-time portfolio updates
- Instant alert delivery
- Live collaboration features

Deployment (Good to have, though not mandatory):

- Dockerized application
- Cloud-hosted (student-friendly platforms)
- Basic real-time data handling
- Simple scaling and monitoring

Compliance & Security (Good to have, though not mandatory):

- Basic data protection practices
- Simulated financial data (no real trading)
- Educational use disclaimer
- Simple data encryption

Success Metrics (Good to have, though not mandatory):

- Prediction accuracy rates
- User engagement and retention
- Portfolio performance improvement
- System uptime and latency

6. Virtual Conference Translator & Summarizer

Objective: Enable real-time multilingual captioning and summarization during live events using speech-to-text, translation, and summarization models for enhanced accessibility and comprehension.

Key Requirements:

Frontend:

- React interface with multilingual live captions
- Real-time Q&A panel and chat integration
- Summary dashboard with key points
- Language selection and preference settings

Backend:

- Node.js with real-time audio processing
- PostgreSQL for session data and user preferences
- Whisper or OpenAI ASR for speech-to-text
- Translation and summarization pipelines

Authentication:

- JWT tokens for session management
- OAuth integration (Google/GitHub) for easy login
- Basic role management (viewer, moderator, host)

AI Integration:

- Whisper for accurate speech transcription
- GPT-4 or Claude 3.5 for content summarization
- Translation models for multilingual support
- Context-aware clarification generation

Agents:

- Audio transcription agent
- Real-time translation agent
- Content summarization agent
- Glossary and terminology agent

Real-Time (Good to have, though not mandatory):

- WebSockets for live caption streaming
- Audio stream synchronization
- Real-time translation updates
- Live summary generation

Deployment (Good to have, though not mandatory):

- Dockerized application
- Cloud-hosted with GPU support for audio processing

- Basic CDN for global reach
- Simple scaling for concurrent sessions

Success Metrics (Good to have, though not mandatory):

- Caption accuracy and latency (<3 seconds)
- Translation quality scores
- User engagement and satisfaction
- Session completion rates

7. Architectural Design Assistant

Objective: A creative tool for architects and engineers that transforms text briefs and rough sketches into refined design concepts using multimodal AI, with compliance checking and design narrative generation.

Key Requirements:

Frontend:

- React interface with drawing tools
- Interactive sketch input and annotation
- 3D design preview and manipulation
- Design iteration and version control

Backend:

- FastAPI with architectural project management
- PostgreSQL for design data and user projects
- Image processing pipeline for sketch analysis
- Integration with design reference databases

Authentication:

- JWT tokens for session management
- OAuth integration (Google/GitHub) for easy login
- Basic project sharing and collaboration features

AI Integration:

- GPT-4V or Gemini for multimodal design analysis
- Stable Diffusion for design concept generation

- Specialized architectural reasoning models
- Code compliance checking AI

Agents (Good to have, though not mandatory):

- Sketch analysis and refinement agent
- Design concept generation agent
- Building code compliance checking agent
- Design narrative and annotation agent

Real-Time (Good to have, though not mandatory):

- SSE for collaborative design sessions
- Live design updates and feedback
- Real-time compliance checking
- Instant design iteration

Deployment (Good to have, though not mandatory):

- Dockerized application with GPU support
- Cloud-hosted with image processing capabilities
- Basic 3D rendering and visualization
- Simple project backup and storage

Success Metrics (Good to have, though not mandatory):

- Design quality and usability scores
- Code compliance accuracy
- User creativity and productivity improvement
- Design iteration efficiency

8. Social Community Moderator & Content Curator

Objective: An automated platform that moderates posts and curates' content for online communities using AI to analyse text and images, with context-aware decision making and community-specific rule enforcement.

Key Requirements:

Frontend:

- React dashboard for moderators and administrators

- Content flagging and review interface
- Community analytics and insights
- User trust and reputation management

Backend:

- Node.js with community content management
- PostgreSQL for posts, users, and moderation logs
- Content analysis and classification pipeline
- Integration with existing community platforms

Authentication:

- JWT tokens for session management
- OAuth integration (Google/GitHub) for easy login
- Role-based access (user, moderator, admin)
- Community-specific permissions

AI Integration:

- GPT-4 or Claude 3.5 for text content analysis
- GPT-4V for image and media scanning
- Sentiment analysis and toxicity detection
- Community-specific rule interpretation

Agents (Good to have, though not mandatory):

- Content classification and analysis agent
- Policy violation detection agent
- Community resource recommendation agent
- User behaviour pattern analysis agent

Real-Time (Good to have, though not mandatory):

- WebSockets for instant moderation alerts
- Live content monitoring and flagging
- Real-time moderator notifications
- Community activity streaming

Deployment (Good to have, though not mandatory):

- Dockerized application
- Cloud-hosted with content processing capabilities

- Basic scaling for community growth
- Simple integration with existing platforms

Success Metrics (Good to have, though not mandatory):

- Moderation accuracy and false positive rates
- Community engagement and health scores
- Response time to policy violations
- User satisfaction and retention

9. Smart Learning Path Generator

Objective: Develop an adaptive educational platform that creates personalized learning paths for students based on their knowledge level, learning style, and goals. The system uses AI to generate custom content, quizzes, and track progress with intelligent tutoring capabilities.

Key Requirements:

Frontend:

- React interface with interactive learning modules
- Progress tracking and analytics dashboard
- Adaptive quiz and assessment system
- Collaborative study group features

Backend:

- Node.js with educational content management
- PostgreSQL for user progress and learning analytics
- Content generation and curation pipeline
- Integration with educational resource APIs

Authentication:

- JWT tokens for session management
- OAuth integration (Google/GitHub) for easy login
- Student and instructor role management
- Progress sharing and parental controls

AI Integration:

- GPT-4 or Claude 3.5 for content generation and tutoring

- Learning analytics and progress assessment
- Personalized difficulty adjustment
- Intelligent question generation and explanation

Agents (Good to have, though not mandatory):

- Learning assessment and profiling agent
- Content generation and curation agent
- Progress tracking and analytics agent
- Personalized tutoring and explanation agent

Real-Time (Good to have, though not mandatory):

- WebSockets for live tutoring sessions
- Real-time progress updates and feedback
- Collaborative learning group chat
- Instant help and clarification system

Deployment (Good to have, though not mandatory):

- Dockerized application
- Cloud-hosted with content delivery optimization
- Basic analytics and performance tracking

Success Metrics (Good to have, though not mandatory):

- Learning outcome improvement rates
- Content quality and relevance scores

10. AI-Powered Recipe Generator & Meal Planner

Objective: Create an intelligent culinary assistant that generates personalized recipes based on dietary preferences, available ingredients, and nutritional goals. The system uses computer vision to identify ingredients from photos and suggests complete meal plans with shopping lists.

Key Requirements:

Frontend:

- React interface with ingredient photo upload
- Recipe discovery and filtering system
- Interactive meal planning calendar

- Shopping list generator and nutrition tracker

Backend:

- Node.js with recipe and nutrition database
- PostgreSQL for user preferences and meal history
- Image processing for ingredient recognition
- Integration with nutrition APIs and databases

Authentication:

- JWT tokens for session management
- OAuth integration (Google/GitHub) for easy login
- Personal profile with dietary preferences and restrictions

AI Integration:

- GPT-4 or Claude 3.5 for recipe generation and adaptation
- Computer vision models for ingredient identification
- Nutritional analysis and meal optimization
- Personalized recommendation system

Agents (Good to have, though not mandatory):

- Ingredient recognition and inventory agent
- Recipe generation and adaptation agent
- Nutritional analysis and optimization agent
- Meal planning and scheduling agent

Real-Time (Good to have, though not mandatory):

- WebSockets for live cooking assistance
- Real-time ingredient substitution suggestions
- Live nutritional calculations
- Collaborative meal planning features

Deployment (Good to have, though not mandatory):

- Dockerized application
- Cloud-hosted with image processing capabilities
- Basic CDN for recipe images and media
- Simple scaling for user growth

Success Metrics (Good to have, though not mandatory):

- Recipe quality and user satisfaction scores
- Ingredient recognition accuracy
- Nutritional goal achievement

Project Considerations:

- **API Development:** Demonstrate your ability to design and implement efficient, scalable, and secure APIs that facilitate seamless communication between the frontend and backend.
- **Authentication & Authorization:** Ensure secure access to your applications by implementing robust authentication (e.g., JWT, OAuth 2.0) and authorization mechanisms, including role-based access control (RBAC) where applicable.
- **Notification Management:** Integrate systems to manage and deliver timely notifications to users, enhancing user engagement and experience.
- **AI Integration:** Effectively incorporate generative AI models to solve the given problem, showcasing an understanding of AI's capabilities and limitations.
- **Documentation:** Provide clear and comprehensive documentation detailing the setup process, usage instructions, and any external services or APIs integrated into your application.

Recommended Tech Stack for Full Stack Developer Assignments:

Frontend Development:

- **JavaScript Frameworks:** React, Angular, or Vue.js for building dynamic and responsive user interfaces.
- **CSS Frameworks:** Bootstrap, Tailwind CSS, or Material-UI for styling and accelerating the development process.
- **Web Components:** Consider using web components (e.g., LitElement, StencilJS) for creating reusable UI components, especially if aiming for a micro-frontend architecture.

Backend Development:

- **Programming Languages:** Python (Flask, Django, FastAPI) or Node.js (Express, NestJS) for backend development, given their robust support for web development and AI integration.
- **APIs:** Develop RESTful or GraphQL APIs for efficient frontend-backend communication.

- **Database:**
    - **SQL Databases:** PostgreSQL or MySQL for relational data storage needs.
    - **NoSQL Databases**: MongoDB or Firebase for flexible, schema-less data structures or real-time data handling.

Authentication & Authorization:

- **JWT (JSON Web Tokens):** For securely handling user authentication and session management.
- **OAuth 2.0:** Implement OAuth for third-party authentication, useful if your project involves sign-in with services like Google, Facebook, etc.

AI and Machine Learning Integration:

- **Hugging Face Transformers:** Access a wide range of pre-trained models for text and image generation tasks.
- **LLM API:** Utilize models like GPT-4 or Grok or gemini for natural language processing and generation tasks.
- **Generative AI Models:** Integrate with models such as DALL-E for image generation or other relevant AI tools as needed.

Cloud Deployment:

- **Platforms:** AWS, Google Cloud Platform (GCP), Azure, or Heroku for deploying and scaling your applications.
- **Containerization:** Use Docker for containerizing applications to ensure consistency across different environments.

Submission Guidelines:

Your submission should include a GitHub repository link containing all source code, a README file with setup instructions, and a brief overview of your project.

Please ensure your project is publicly accessible or share a private link with access permissions for our review team.

Please ensure the project you submit is solely your work and is not borrowed / picked up from internet sources.

Evaluation Criteria:

- **Creativity**: Demonstrate innovative approaches to problem-solving and the creative use of AI and full-stack technologies.
- **Technical Proficiency**: Showcase effective use of relevant technologies, frameworks, and programming techniques.
- **Functionality:** Ensure that the solution meets all project requirements and functions as intended.
- **Time Management**: Ability to deliver a functional and polished project within the 4-day timeframe.
- **Documentation:** Provide clear, thorough, and well-organized documentation that facilitates understanding and deployment of your application.

Additional Information:

- **Data Sources:** Feel free to use any public data sources as needed. Ensure to respect data privacy and usage rights.
- **Free Tiers:** We recommend using the free tiers of cloud services and AI model APIs for this project. Keep an eye on usage to avoid incurring costs.

We are excited to see how you approach these challenges and look forward to your innovative solutions. This is your opportunity to showcase your full-stack development skills and your ability to integrate generative AI technologies effectively.

Best regards,

Nebula9.ai Team

Business Analyst Internship Assignment - Nebula9.ai

Dear Students,

Thank you for your interest in the Business Analyst position at Nebula9.ai. We are excited at the prospect of you bringing your analytical prowess and strategic thinking to our team. Our mission is to harness the power of generative AI to solve real-world problems, and your role would be crucial in identifying opportunities and optimizing our offerings.

As part of our selection process, we have designed an assignment to evaluate your ability to analyze data, understand business implications, and provide actionable insights. This project will give us a glimpse into your problem-solving skills and your approach to business analysis in the tech-driven landscape of generative AI.

List of Assignments:

1. **Scenario: AI-Powered Retail Inventory Management**

You have been assigned as the Business Analyst for a retail company that wants to use Generative AI to automate inventory management. The system should predict stock levels based on sales trends, optimize order quantities, and reduce wastage by improving demand forecasting.

2. **Scenario: AI in Healthcare Patient Support**

You have been assigned as the Business Analyst for a healthcare provider seeking to implement an AI-powered virtual assistant to provide patients with 24/7 support. The assistant should help patients schedule appointments, answer common health queries, and assist with medication reminders while integrating with the hospital's electronic medical records (EMR) system.

3. **Scenario: AI for Smart Manufacturing Quality Control**

You have been assigned as the Business Analyst for a manufacturing company that is looking to implement an AI-driven quality control system. The solution should analyze production data in real time, detect defects in products, and suggest corrective actions to reduce wastage and downtime.

4. **Scenario: AI-Powered Marketing Campaign Optimization**

You have been assigned as the Business Analyst for a digital marketing agency that wants to implement a Generative AI solution to optimize marketing campaigns. The system should analyze past campaign performance, suggest improvements, and automate content creation based on customer engagement data.

5. **Scenario: AI in E-Learning for Personalized Learning Experiences**

You have been assigned as the Business Analyst for an e-learning platform that wants to implement AI to personalize learning experiences. The system should recommend courses, adapt learning paths based on student performance, and offer real-time tutoring assistance by analyzing students' engagement and progress.

6. **Scenario: AI for Sports Performance Analysis**

You have been assigned as the Business Analyst for a sports training organization that wants to use AI for performance analysis. The AI system should analyze players’ performance during training and matches, providing insights for improvement, injury prevention, and personalized training programs.

7. **Scenario: AI-Driven Financial Risk Management**

You have been assigned as the Business Analyst for a financial services firm that is looking to leverage AI to improve risk management. The system should identify high-risk clients or transactions, predict market volatility, and provide insights on mitigating financial risks.

8. **Scenario: AI-Powered Supply Chain Optimization**

You have been assigned as the Business Analyst for a logistics company that is exploring AI solutions to optimize its supply chain. The AI system should predict delivery times, optimize routes for cost efficiency, and manage warehouse inventories across multiple locations.

9. **Scenario: AI-Powered Sports Fan Engagement**

You have been assigned as the Business Analyst for a sports team that wants to use Generative AI to enhance fan engagement. The system should create personalized content for fans, predict match

outcomes, and optimize fan interactions during live events through real-time AI-driven insights and recommendations.

10. **Scenario: AI for Sustainable Agriculture**

You have been assigned as the Business Analyst for an agricultural firm that is looking to use AI to improve crop yields and sustainability. The AI system should analyze environmental data (e.g., weather, soil conditions) to optimize irrigation and fertilization, predict yields, and reduce the environmental impact of farming practices.

—

Objective: Your task is to complete a business analysis project based on the allocated scenario.

Project Task:

You are tasked with developing a complete business case, project plan, and solution overview for the implementation of the AI-powered system relevant to the assigned scenario.

Deliverables:

1. Project Management Plan (PM Capability)

Task: Develop a streamlined project plan for the AI-powered solution in the assigned scenario.
Deliverables:

- A simple Gantt chart outlining key project phases (e.g., discovery, design, development, testing, deployment).
- A brief risk management plan highlighting potential risks, mitigation strategies, and resource considerations.

---

2. Business Case Presentation & Communication

Task: Create and communicate a business case for the AI solution.
Deliverables:

- A 10-slide presentation (using PowerPoint, Google Slides, or Canva) including:
    - Executive Summary
    - Problem & Proposed Solution
    - Key Benefits & Cost Overview
    - Conclusion & Business Impact
- A 1-page explanation of the AI solution in **non-technical language**, written for a business stakeholder (e.g., COO, Head of Operations), emphasizing business value and strategic fit.
- *Optional:* Record a short 2–5 minute video or role-play explaining the solution to a non-technical audience.

---

3. Feasibility & Solution Analysis (Logical Thinking)

Task: Evaluate the feasibility of implementing the AI system, considering technical and business constraints.
Deliverables:

- A 1–2 page feasibility report outlining implementation advantages, expected challenges, barriers, and a basic timeline.
- A **decision matrix** comparing 2–3 relevant AI models or approaches (e.g., LangChain, AutoGen, GPT-4, Claude, RAG frameworks) based on cost, scalability, accuracy, and integration complexity.

---

4. Innovation in AI (Creative Thinking)

Task: Propose a unique AI feature or idea that can differentiate the solution in your scenario.
Deliverables:

- A 1-page document describing the feature, its purpose, and its potential impact.
- A simple diagram or workflow to visually represent how this feature works within the AI solution.

Deliverables Summary:

Project Management Plan

- Gantt chart outlining project phases such as discovery, design, development, testing, and deployment.
- Risk management plan detailing potential risks, mitigation strategies, and resource allocation.

Business Case & Communication

- 10-slide presentation covering the executive summary, problem statement, proposed AI solution, key benefits, cost estimates, and ROI projections.
- 1-page non-technical summary for business stakeholders, clearly explaining the solution and its business value**.**
- Optional: A 5-minute recorded video or a role-play presentation explaining the solution to a non-technical audience.

Feasibility & Solution Analysis

- 2-page report analyzing the implementation feasibility, including technical and business factors, expected challenges, advantages, and a proposed timeline.
- A decision matrix comparing relevant AI models or approaches based on factors such as cost, scalability, accuracy, and ease of integration.

Innovation in AI

- 1-page proposal outlining a unique and value-driven AI feature, its functionality, and how it enhances the solution.

- A diagram or workflow visually representing the proposed feature's function within the system.

Evaluation Criteria:

- Project Management: Planning and risk management.
- Communication: Clarity and effectiveness.
- Logical Thinking: Analysis and decision-making.
- Creative Thinking: Originality and practicality.

Submission Guidelines:

Please submit your report and presentation through the Google Form shared in the email. Ensure your submission is concise, well-structured, and communicates your analysis clearly.

We are eager to explore your insights!

Best of luck,

Nebula9.ai