Nutrition Database

Project Name: NutriPro

Group Number: 002

Team members

Student No.	Full Name	GitHub Username	Contribution (sum to 100%)
s2817538	Kate Baker	MadH3r3K8	33.3% or Equal
s5374957	I Wibawa	IWibawa	33.3% or Equal
s5414931	Naveen Arakkal Nelson	narakkal-nelson	33.3% or Equal

Brief Description of Contribution

- s2817538, Kate Baker
  - Accomplishments: Project Plan

1. **Defining Project Objectives**: Clearly outline the goals of the nutritional database. This might include providing accurate nutritional information, supporting dietary planning, or integrating with other health apps.
2. **Scope Definition**: Establish the boundaries of the project. Determine what types of nutritional data will be included (e.g., macronutrients, micronutrients, allergens), and what features the database will support (e.g., search functionality, user input).
3. **Data Collection and Validation**: Identify reliable sources for nutritional data and establish methods for validating this information. This ensures the database contains accurate and trustworthy data.
4. **Task Breakdown**: Create a Work Breakdown Structure (WBS) that details all tasks required to develop the database. This might include data collection, database design, user interface development, and testing.
5. **Resource Allocation**: Assign the necessary resources, such as data analysts, database developers, and nutrition experts. Ensure that the project has the required budget, tools, and technology.
6. **Timeline Creation**: Develop a detailed schedule that outlines when each task will be completed. Set milestones for key phases such as data collection, database development, and user testing.
7. **Risk Management**: Identify potential risks, such as data inaccuracies or technical challenges, and develop strategies to mitigate these risks. This helps to anticipate and address issues proactively.

8. **Communication Plan**: Establish how information will be shared among team members and stakeholders. Set up regular meetings, progress reports, and communication channels to ensure everyone is informed and aligned.
9. **Quality Assurance**: Define the quality standards for the nutritional data and the database functionality. Develop testing protocols to ensure the database meets these standards before launch.
10. **Stakeholder Engagement**: Identify all stakeholders, including end-users, nutrition experts, and regulatory bodies. Understand their needs and expectations to ensure the database meets their requirements.
11. **Approval and Sign-off**: Obtain formal approval from stakeholders on the project plan. This ensures that everyone agrees with the plan and is committed to its success.

- s5374957, I Wibawa
    - Accomplishments: Software Development

1. **Defining Project Objectives**: Clearly outline the goals of the software. This includes providing accurate nutritional information, supporting dietary planning, and enabling nutritional data visualisation as specified in the SDD.
2. **Scope Definition**: Establish the boundaries of the project. Determine the types of nutritional data to be included (e.g., macronutrients, micronutrients, vitamins, minerals) and the features the software will support (e.g., food search, nutrition breakdown visualisation, filtering, comparison, and meal planning).
3. **System Architecture Design**: Develop a high-level design of the software architecture. This includes deciding on the technology stack, database structure, and defining the User Interface, Application Layer, and Data Layer components as outlined in the SDD.
4. **Data Collection and Validation**: Utilise the specified Nutritional_Food_Database.csv file and establish methods for validating this information. This ensures the database contains accurate and trustworthy data.
5. **Task Breakdown**: Create a Work Breakdown Structure (WBS) that details all tasks required to develop the software. This includes data processing, user interface development, implementing search functionality, creating visualisation components, and developing the meal planning feature.
6. **Resource Allocation**: Assign the necessary resources, such as software developers and data analysts. Ensure that the project has the required budget, tools, and technology to implement all specified components and features.
7. **Timeline Creation**: Develop a detailed schedule that outlines when each task will be completed. Set milestones for key phases such as database development, UI implementation, and user testing.
8. **Risk Management**: Identify potential risks, such as data inaccuracies, technical challenges, or security vulnerabilities, and develop strategies to mitigate these risks. This helps to anticipate and address issues proactively.
9. **Communication Plan**: Establish how information will be shared among team members and stakeholders. Set up regular meetings, progress reports, and communication channels to ensure everyone is informed and aligned.
10. **Quality Assurance**: Define the quality standards for the software and the nutritional data. Develop testing protocols to ensure the software meets these standards before launch. This includes unit testing, integration testing, user acceptance testing, and specific tests for requirements like response time and calculation accuracy.
11. **User Interface and Experience Design**: Design an intuitive and user-friendly interface based on the structural and visual designs outlined in the SDD. Ensure that the software is easy to navigate and meets

the needs of its users.

12. **Stakeholder Engagement**: Identify all stakeholders, including end-users (health-conscious individuals, dieters, nutritionists, and dietitians), nutrition experts, and regulatory bodies. Understand their needs and expectations to ensure the software meets their requirements.
13. **Data Structures and Functions Implementation**: Develop and implement the core data structures (FoodItem, NutritionalDatabase, Chart, ComparisonResult) and functions as specified in the SDD.
14. **Additional Feature Development**: Plan and implement the additional feature mentioned in the SDD, beyond the required features.
15. **Documentation**: Maintain and update the Software Development Document throughout the development process, ensuring it remains a comprehensive and accurate representation of the system.
16. **Approval and Sign-off**: Obtain formal approval from stakeholders on the project plan. This ensures that everyone agrees with the plan and is committed to its success.

- s5414931, Naveen Arakkal Nelson
  - Accomplishments: Gant Chart

1. **Clear Project Timeline:** Establish Start and End Dates: Define the overall project duration. Set Milestones: Identify key dates for major deliverables or phases.
2. **Task Breakdown: Identify Tasks**: List all tasks required to complete the project. Define Task Dependencies: Determine which tasks depend on the completion of others.
3. **Resource Allocation:** Assign Responsibilities: Allocate tasks to team members. Balance Workload: Ensure that no team member is overloaded with tasks.
4. **Progress Tracking:** Set Deadlines: Assign specific deadlines for each task. Monitor Progress: Track the completion status of tasks in real-time.
5. **Visual Representation:** Create Visual Timeline: Use bars to represent the duration of each task. Highlight Critical Path: Identify the sequence of tasks that directly impact the project timeline.
6. **Communication Tool:** Share with Stakeholders: Provide a clear visual representation of the project timeline to all stakeholders. Update Regularly: Keep the Gantt chart updated to reflect any changes in the project plan.

Table of Contents

- Project Plan
    - 1. Project Overview
        - 1.1 Project Objectives
        - 1.2 Project Stakeholders
        - 1.3 Project Scope
    - 2. Work Breakdown Structure
    - 3. Activity Definition Estimation
    - 4. Gantt Chart

1. Project Overview

1.1 Project Objectives

NutriPro is used for a variety of purposes including standards development, nutrition labelling, research on diet and disease, education and to help consumers make better informed food choices. the main project objectives are:

- To serve the requirments of the potential users
- Work cost-effective and with timeframe for editions
- Consult with all stakeholder and interested parties and users
- Provided continues access of all users to the outputs
- Disseminate program widely

Nutirital Database Programs Objectives

1. Name: Food Search Description: Enable users to search for foods by name and display all the nutritional information;
2. Name: Nutrition Breakdown Description: Enable users to select one food, and display pie charts & bar graphs showing the breakdown of different nutrients for the selected food.
3. Name: Nutrition Range Filter Description: Enable users to select one of nutrition and input minimum & maximum values, and the tool will display a list of foods that fall within those ranges.
4. Name: Nutrition Level Filter Description: Enable users to filter foods by nutritional content levels—low, mid, and high—including fat, protein, carbohydrates, sugar, and nutritional density. The three levels are defined as follows: Low: Less than 33% of the highest value. Mid: Between 33% and 66% of the highest value. High: Greater than 66% of the highest value.
5. Standard Procedures are in line with international guildlines
6. Integrated in an international network
7. Document all data
8. Systematically be able to explain your data and data choices.
9. Laugauge (English, Chinese, Japanese)
10. Food Identification (classification/description)
11. Food Numbering (indicating food groups and swquential numbering within)

1.2 Project Stakeholders

In a NutriPro database, stakeholders play crucial roles in ensuring the effective collection, management, and utilization of nutrition-related data. Here are some key stakeholders typically involved:

Internal Stakeholder

1. **Company Executives**:Chief Executive Officers, Chief Finacial Officers, Chief Cyber Security Officer, Chief Operating Office, Chief Marketing Officer, Chief Technology Officer, Chief Information Officer,

General Council, Vice President
2. **IT Team**: IT Manager, System Administator, Network Engineer, IT Support Specialist, Software Developer, Data Administrator, Security Analyst, QA Tester, IT Project Manager, Data Analyst
3. **Marketing Team**: Marketing Manager, SEO Specialist, Content Marketer, Data Analyst, Brand Strategist, Public Relations Manager, Visual Designer, Social Media Manager & Website Developer
4. **Legal Team**: Partners, Associates, Paralegals, Legal Assistance, General Counsel, Contract Managers & Compliance Officers
5. **Customer Service team**: Customer Service Representative, Customer Service Manager, Technical Support Specialist, Quality Assuarance Analyst, Customer Success Manager, Training and Development Specialist, Social Media Manager, Live Chat Support Agent & Customer Experiance Specialist.

External Stakeholders

1. **Customers/Cliental/User**: These stakeholders use the data to monitor and assess the impact for their food intake. Allow to moniotor their daily calorie count.
2. **Government Agencies**: These include health departments, agricultural ministries, and education departments that use the data for policy-making and program implementation.
3. **Non-Governmental Organizations (NGOs)**: NGOs often contribute data, help in data collection, and use the database to plan and monitor their nutrition programs.
4. **Healthcare Providers**: Hospitals, clinics, and community health workers provide essential data on nutritional status and outcomes.
5. **Academic and Research Institutions**: These stakeholders analyze the data to conduct research and provide evidence-based recommendations.
6. **International Organizations**: Entities like the World Health Organization (WHO) and the United Nations Children's Fund (UNICEF) use the data for global nutrition monitoring and support.
7. **Private Sector**: Companies involved in food production and distribution may use the data to improve product offerings and ensure they meet nutritional standards.
8. **Community Groups**: Local organizations and community leaders help in data collection and dissemination of information to the public.
9. **Donors and Funding Agencies**: These stakeholders use the data to assess the impact of their investments and guide future funding decisions.

These stakeholders collaborate to ensure the database is comprehensive, accurate, and useful for improving nutrition outcomes.

1.3 Project Scope

A NutriPro Database is a comprehensive collection of data on the nutritional content of various foods. It typically includes information on macronutrients (like carbohydrates, proteins, and fats), micronutrients (such as vitamins and minerals), and other components like fiber and water content.

What is Included:

1. **Food Items**: A wide range of foods, including raw ingredients, processed foods, and restaurant meals.
2. **Nutrient Information**: Detailed data on macronutrients, micronutrients, and other dietary components.

3. **Serving Sizes**: Standardized serving sizes to ensure consistency.
4. **Food Categories**: Classification of foods into categories like fruits, vegetables, grains, etc.
5. **Source Information**: Details about the source of the data, such as laboratory analysis or manufacturer-provided information.

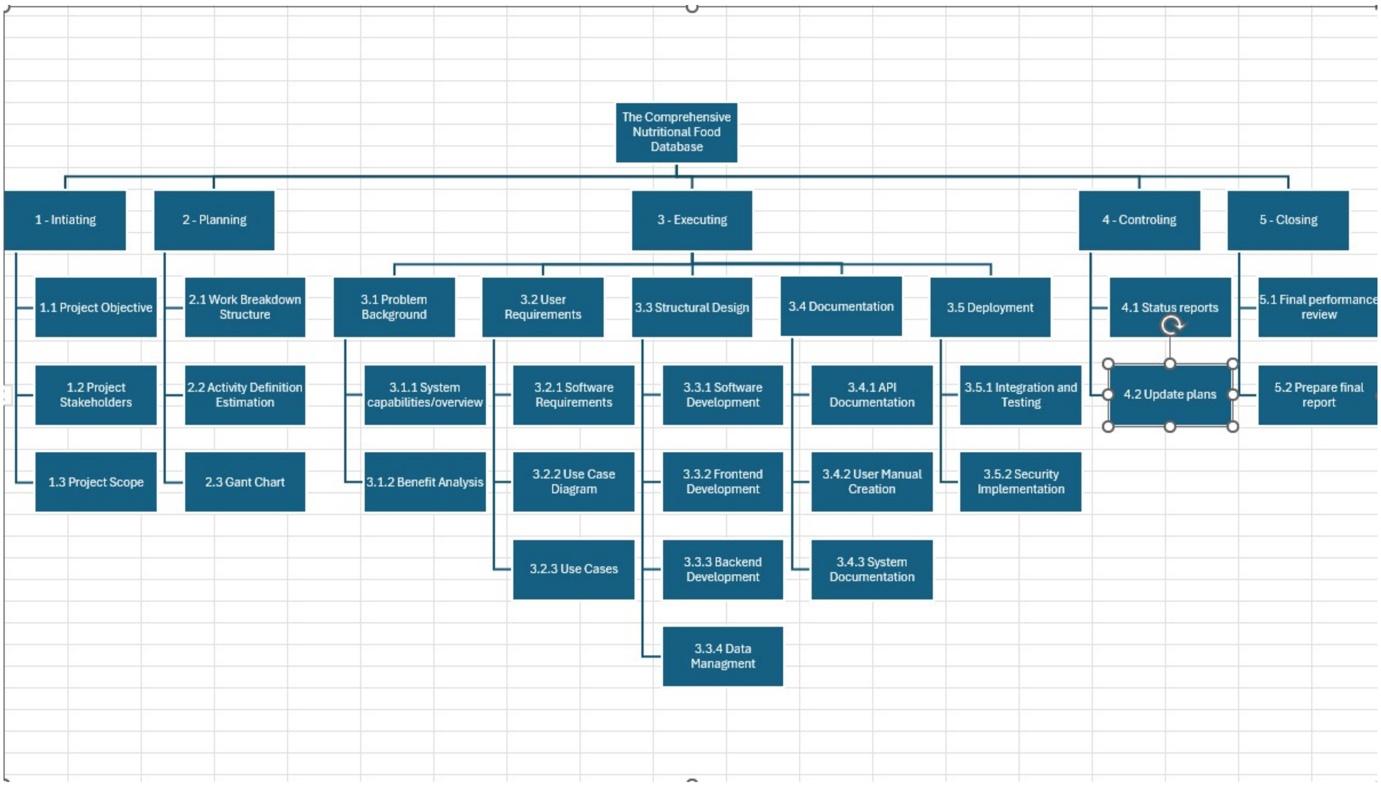
Managing What is Included:

1. **Data Collection**: Gathering data from reliable sources, including scientific research, food manufacturers, and government databases.
2. **Data Verification**: Ensuring the accuracy and reliability of the data through rigorous validation processes.
3. **Regular Updates**: Continuously updating the database to reflect new research, changes in food formulations, and new food products.
4. **User Feedback**: Incorporating feedback from users to improve the database's accuracy and usability.

Excluding Data:

1. **Inaccurate Data**: Excluding data that cannot be verified or is found to be inaccurate.
2. **Outdated Information**: Removing outdated information that no longer reflects current food compositions.
3. **Non-Nutritional Data**: Excluding data that is not relevant to nutritional content, such as marketing claims or non-nutritional additives.

2. Work Breakdown Structure



3. Activity Definition Estimation

Define the activities required for your project based on the WBS, and assign responsibilities to team members. Each activity should be numbered and correspond with your Gantt chart. Provide estimated durations for each activity to facilitate Gantt chart preparation.

Activity #	Activity Name	Brief Description	Duration	Responsible Team Members
1. **Initiating**	1.1 Project Objective	1. **Understand the Project Scope**: Clearly define the boundaries and deliverables of the project. This helps in setting realistic objectives. 2. **Identify Stakeholder Needs**: Gather input from all stakeholders to understand their expectations and requirements. 3. **Set SMART Objectives**: Ensure that the objectives are Specific, Measurable, Achievable, Relevant, and Time-bound. 4. **Align with Business Goals**: Make sure the project objectives align with the broader business goals and strategies. 5. **Document Objectives**: Write down the objectives clearly and concisely. This helps in communicating them effectively to the project team and stakeholders. 6. **Review and Validate**: Get feedback from stakeholders and make necessary	**(5 Days)**	**All Team Members** Kate as Lead Chief Officer

		adjustments to ensure the objectives are realistic and agreed upon.		
	1.2 Project Stakeholders	1. **Identify Stakeholders**: Determine who the stakeholders are. This includes anyone who has an interest in or is affected by the project. 2. **Analyze Stakeholders**: Understand each stakeholder’s interests, influence, and impact on the project. This helps prioritize their needs and manage their expectations. 3. **Engage Stakeholders**: Develop strategies to engage stakeholders based on their level of interest and influence. This might include regular meetings, updates, and feedback sessions. 4. **Manage Stakeholder Expectations**: Continuously communicate with stakeholders to manage their expectations and address any concerns. This helps build trust and ensures their support throughout the project. 5. **Monitor and Review**: Regularly review stakeholder engagement and adjust as needed to ensure ongoing alignment with project goals	**(14 Days)**	**All Team Members** Kate as Lead Chief Officer

	1.3 Project Scope	1. **Identify Project Objectives**: Clearly define what the project aims to achieve. This includes setting specific, measurable, achievable, relevant, and time-bound (SMART) goals. 2. **Gather Requirements**: Collect all the necessary requirements from stakeholders to understand what needs to be delivered. 3. **Define Deliverables**: List all the deliverables that the project will produce. This helps in understanding the scope of work. 4. **Create a Work Breakdown Structure (WBS)**: Break down the project into smaller, manageable tasks and activities. 5. **Identify Exclusions**: Clearly state what is not included in the project to avoid any misunderstandings later. 6. **Establish Constraints**: Identify any limitations or constraints such as budget, time, and resources. 7. **Assumptions**: Document any assumptions that are made during the planning phase. 8. **Get Stakeholder Approval**: Ensure that all key stakeholders review and approve the project scope	**(5 Days)**	**All Team Members** Kate as Lead Chief Officer

		to confirm alignment and understanding.		
2 - Planning	2.1 - Work Breakdown Structure	1. **Hierarchy**: The WBS is structured in a hierarchical manner, starting with the overall project at the top and breaking it down into smaller deliverables and tasks. 2. **Deliverables**: Each level of the WBS represents a more detailed breakdown of the project deliverables. 3. **Phases**: The project can be divided into phases, with each phase containing specific tasks and deliverables. 4. **100% Rule**: This rule ensures that the WBS includes all the work required for the project, with no overlap or omission.	**(4 Days)** Continues Updating & End of Section Updates	**All Team Members** Kate as Lead Chief Officer
	2.2 - Activity Definition Estimation	1. **Purpose**: To determine the amount of time needed for each activity, which helps in developing the project schedule and estimating costs. 2. **Factors**: Several factors influence these estimates, including the complexity of the activity, resource availability, team expertise, risk assessment, and external dependencies[1]. 3. **Techniques**: Common techniques include:	**(5 days)** Continues Updating & End of Section Updates	**All Team Members** Neveen as Lead Chief Officer

		◦ **Expert Judgment**: Leveraging the experience of experts to estimate durations. ◦ **Analogous Estimating**: Using historical data from similar projects. ◦ **Parametric Estimating**: Applying statistical relationships between historical data and other variables. ◦ **Three-Point Estimating**: Considering optimistic, pessimistic, and most likely scenarios to calculate an average duration. ◦ **Bottom-Up Estimating**: Breaking down activities into smaller components and estimating each one		
	2.3 - Gant Chart	1. **Tasks and Time**: Tasks are listed on the vertical axis, and time intervals are shown on the horizontal axis. 2. **Bars**: Each bar represents a task, with the length of the bar indicating the duration of the task. 3. **Dependencies**: Modern Gantt charts can show dependencies between tasks, indicating which tasks need to be completed before others can start.	**(4 days)** Continues Updating & End of Section Updates	**All Team Members** Naveen as Lead Chief Officer

		4. **Progress Tracking**: They often include features to track the progress of tasks, such as shading or color-coding.		
3- Executing	3.1 – Problem Background	**System capabilities/overview** 1. **Identify Stakeholder Needs**: Gather and analyze the needs and requirements of all stakeholders to understand what the system must accomplish. 2. **Define System Objectives**: Clearly outline the goals and objectives of the system, ensuring they align with stakeholder needs and organizational goals. 3. **Develop System Requirements**: Translate stakeholder needs into detailed system requirements, specifying what the system should do and how it should perform. 4. **Create a System Architecture**: Design the overall structure of the system, including its components, their relationships, and how they interact. 5. **Define Capabilities**: Specify the key capabilities of the system, detailing the functions it will perform and the performance criteria it must meet. 6. **Develop a Concept of Operations (CONOPS)**: Describe how the system	**(3 days)** Continues Updating & End of Section Updates **(5 days)** Continues Updating & End of Section Updates	**All Team Members** Kate as Lead Chief Officer

		will be used in real-world scenarios, including its operational environment and user interactions. 7. **Validate and Verify Requirements**: Ensure that the system requirements are complete, feasible, and testable. This involves reviewing and validating requirements with stakeholders. 8. **Document the Overview**: Compile all the information into a comprehensive document that provides a clear overview of the system’s capabilities, architecture, and operational concept. **Benefits and Analysis** 1. **Establish a Framework**: Define the scope and objectives of the analysis. This includes identifying what you aim to achieve and the criteria for success. 2. **Identify Costs and Benefits**: List all potential costs and benefits associated with the project or decision. This includes both direct and indirect impacts. 3. **Assign Values**: Quantify the costs and benefits in monetary terms where possible. This helps in comparing them on a common scale.			

		4. **Compare Costs and Benefits**: Calculate the total value of the benefits and subtract the total costs. This can be done using various methods such as net present value (NPV) or cost-benefit ratio. 5. **Analyze Results**: Interpret the results to determine whether the benefits outweigh the costs. Consider both quantitative and qualitative factors. 6. **Make Recommendations**: Based on the analysis, make informed recommendations on whether to proceed with the project or decision.		
	3.2 - User Requirements	**Use Case Diagram** 1. **Identify Actors**: Determine the external entities that will interact with the system. These can be users, other systems, or hardware devices. 2. **Identify Use Cases**: Define the specific functions or actions that the system will perform. Each use case represents a goal that an actor wants to achieve. 3. **Establish Relationships**: Determine the relationships between actors and use cases. This includes identifying which actors are involved in which use cases and how they interact.	**(3 days)** Continues Updating & End of Section Updates **(2 days)** Continues Updating & End of Section Updates	**All Team Members** I Wibawa as Lead Chief Officer

4. **Draw the Diagram**: Use a UML (Unified Modeling Language) tool or draw by hand to create the diagram. Place actors outside the system boundary and use cases inside. Connect actors to their respective use cases with lines.
5. **Review and Validate**: Ensure that the diagram accurately represents the system's functionality and interactions. Review it with stakeholders to confirm its accuracy and completeness.

Use Case

1. **Identify Actors**: Determine who will interact with the system. Actors can be users, other systems, or devices.
2. **Define Use Cases**: Identify the specific actions or functions that the system will perform. Each use case represents a goal that an actor wants to achieve.
3. **Establish Relationships**: Determine how actors and use cases interact. This includes identifying which actors are involved in which use cases and how they interact[3].
4. **Outline Preconditions**: Specify any conditions that must be met before the use case can be initiated[24].
5. **Describe the Basic Flow**: Detail the standard

		sequence of steps that occur when the use case is executed successfully. 6. **Identify Alternate Flows**: Outline any variations or exceptions to the basic flow, including error conditions and alternative paths. 7. **Document Postconditions**: Define the state of the system after the use case has been completed. 8. **Review and Validate**: Ensure that the use case accurately represents the system's functionality and interactions. Review it with stakeholders to confirm its accuracy and completeness.			
	3.3 - Structural Design	**Software Development** 1. System architecture design 2. Database schema design 3. UI/UX wireframing 4. Design review and approval **Frontend Development** 1. Set up frontend framework 2. Implement user interface components 3. Develop search and results display 4. Create visualization components (charts and graphs) 5. Implement comparison view 6. Develop meal plan generator interface	**10 days** (3 days) (2 days) (3 days) (2 days) **25 Days** (2 days) (5 days) (5 days) (6 days) (4 days) (3 days)	**All Team Members** I Wibawa as Lead Chief Officer	

		Backend Development 1. Set up development environment 2. Implement data loading from CSV 3. Develop search functionality 4. Implement filtering and categorization logic 5. Create API endpoints for frontend communication **Data Management** 1. Data cleaning and preprocessing 2. Database setup and configuration 3. Data import and validation 4. Implement data update mechanism	**20 Days** (1 day) (3 days) (5 days) (7 days) (4 days) **8 Days** (3 days) (1 day) (3 days) (1 day)	
	3.4 - Documentations	**API Documentations** 1. **Create an Outline**: Start by developing a detailed outline of what the documentation will cover. This helps organize the content and ensures all necessary topics are included. 2. **Select a Platform**: Choose a platform or tool for hosting your documentation. This could be a dedicated documentation site, a wiki, or a platform like GitHub. 3. **Write the Introduction**: Provide an overview of the API, including its purpose, key features, and how it can be used.	**(10 Days)** **(15 Days)** **(12 Days)**	**All Team Members** Kate as Lead Chief Officer

		4. **Create Samples and Tutorials**: Include code samples and tutorials to help users understand how to implement the API in various scenarios. 5. **Explain Authentication and Endpoints**: Detail the authentication methods required to access the API and describe each endpoint, including the methods, parameters, and expected responses. 6. **Write the Reference Material**: Provide detailed reference material for each endpoint, including descriptions of parameters, request and response formats, and error codes. 7. **Troubleshoot**: Include a troubleshooting section to help users resolve common issues they might encounter. 8. **Clean Up**: Review and edit the documentation to ensure clarity, accuracy, and consistency. **User Manual Documentations** 1. **Define Your Audience**: Understand who will be using the manual. This helps tailor the content to their needs and knowledge level. 2. **Build an Outline**: Create a detailed outline of the manual, including sections such as introduction, table of contents, step-by-step		

		instructions, troubleshooting, and FAQs. 3. **Gather Information**: Collect all necessary information about the product or service, including features, functions, and common issues. 4. **Write Clear Instructions**: Use simple, concise language to write step-by-step instructions. Avoid jargon and ensure each step is easy to follow. 5. **Include Visuals**: Add annotated screenshots, diagrams, and other visuals to help users understand the instructions better[12]. 6. **Test the Manual**: Have a few users test the manual to ensure the instructions are clear and accurate. Gather feedback and make necessary adjustments. 7. **Review and Edit**: Proofread the manual for any errors and ensure consistency in formatting and style. 8. **Publish and Distribute**: Make the manual available to users, either in print or online. Ensure it is easily accessible. 9. **Maintain and Update**: Regularly update the manual to reflect any changes or new features in the product or service **System Documentation**			

		1. **Define the Purpose and Scope**: Clearly outline the objectives of the documentation and what it will cover. This helps set expectations and ensures all necessary information is included. 2. **Identify the Audience**: Determine who will be using the documentation, such as developers, users, or maintenance teams. Tailor the content to meet their needs and knowledge levels. 3. **Gather Information**: Collect all relevant information about the system, including its architecture, components, functionalities, and interfaces. 4. **Organize the Content**: Structure the documentation logically, typically starting with an overview and then detailing each component and its interactions. 5. **Write Clear and Concise Content**: Use simple language and avoid jargon. Ensure each section is easy to understand and provides the necessary details. 6. **Include Visuals**: Add diagrams, flowcharts, and screenshots to help illustrate complex concepts and make the		

		documentation more engaging. 7. **Review and Validate**: Have subject matter experts review the documentation to ensure accuracy and completeness. Make necessary revisions based on their feedback. 8. **Publish and Distribute**: Make the documentation accessible to all relevant stakeholders, either in print or online. 9. **Maintain and Update**: Regularly update the documentation to reflect any changes or new features in the system.		
	3.5 - Deployment	**Integration & Testing** 1. Backend-Frontend integration 2. Unit testing 3. Integration testing 4. User acceptance testing 5. Performance testing **Security Implementation** 1. Security requirements analysis (2 days) 2. Implement authentication and authorization (3 days) 3. Data encryption implementation (2 days) 4. Security audit (1 day)	**15 Days** (3 days) (4 days) (3 days) (3 days) (2 days) **8 Days** (2 days) (3 days) (2 days) (1 day)	**All Team Members** I Wibawa as Lead Chief Officer
4 - Controlling	4.1 – Status Report	1. Measuring the ongoing project activities	**30 Days**	**All Team Members**

		2. Monitoring the project variable 3. Identifying corrective actions to address issues and risks properly 4. Influencing the factors that could circumvent integrated change controls	Monitor and Report from a month’s worth of data	Naveen as Lead Chief Officer
	4.2 - Update Plans	1. **Review Current Status**: Assess the current progress of the project against the original plan. Identify any deviations or issues. 2. **Identify Changes**: Determine what changes are needed. This could include scope adjustments, timeline shifts, resource reallocations, or budget updates. 3. **Analyze Impact**: Evaluate the potential impact of the changes on the project. Consider how they will affect the schedule, budget, resources, and overall project goals. 4. **Update Documentation**: Revise the project plan and any related documents to reflect the changes. Ensure all updates are clearly documented. 5. **Communicate Changes**: Inform all stakeholders about the changes. Provide clear explanations and ensure everyone understands the reasons and implications.	**15 Days**	

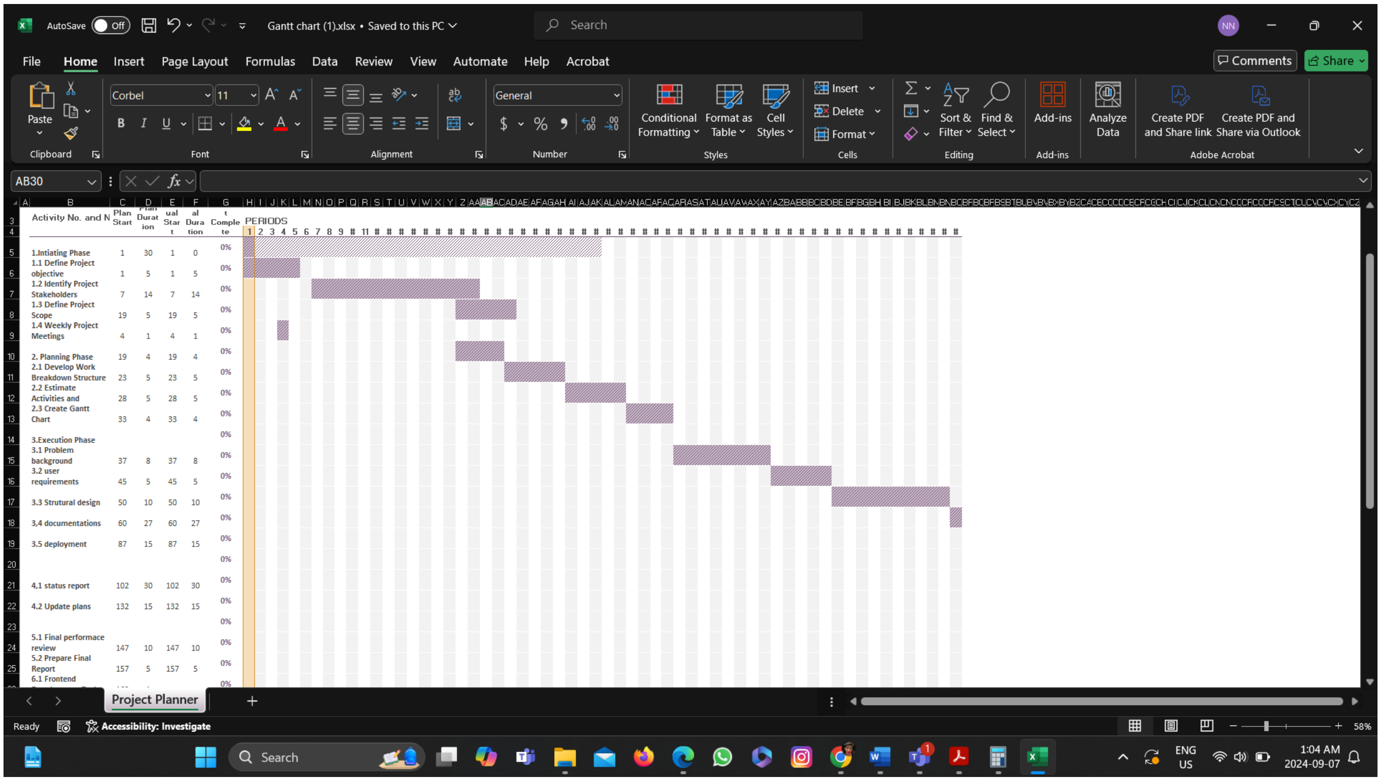
		6. **Implement Changes**: Put the changes into action. Adjust schedules, reassign tasks, and update any tools or systems used to manage the project. 7. **Monitor and Review**: Continuously monitor the project to ensure the changes are having the desired effect. Be prepared to make further adjustments if necessary.		
5 - Closing	5.1 - Final Performance Review	1. **Preparation**: Gather all relevant data on the employee’s performance over the review period. This includes performance metrics, feedback from peers and supervisors, and any self-assessments. 2. **Set the Agenda**: Outline the main topics to be covered during the review, such as achievements, areas for improvement, and future goals. 3. **Create a Positive Environment**: Start the review with a personal greeting and create a comfortable atmosphere for open and honest communication. 4. **Discuss Performance**: Review the employee’s performance against the set objectives and expectations. Highlight both strengths and areas for improvement.	**10 Days**	**All Team Members**

		5. **Provide Feedback**: Offer specific, actionable feedback. Use examples to illustrate points and ensure the feedback is balanced and constructive. 6. **Set Goals**: Collaboratively set goals for the next review period. These should be SMART (Specific, Measurable, Achievable, Relevant, Time-bound) goals. 7. **Document the Review**: Record the key points discussed, including feedback, goals, and any agreed-upon action plans. 8. **Follow-Up**: Schedule follow-up meetings to monitor progress and provide ongoing support and feedback.		
	5.2 - Prepare Final Report	1. **Define the Purpose and Audience**: Understand the purpose of the report and who will be reading it. This helps tailor the content to meet the needs and expectations of the audience. 2. **Gather Information**: Collect all relevant data, findings, and insights from the project. This includes any supporting documents, charts, and graphs. 3. **Create an Outline**: Develop a detailed outline to organize the report logically. Common sections	**5 Days**	**All Team Members**

		include an executive summary, introduction, methodology, results, discussion, conclusion, and recommendations. 4. **Write the Draft**: Start writing the report based on the outline. Ensure each section is clear and concise, and use visuals to support the text where necessary. 5. **Review and Revise**: Review the draft for clarity, coherence, and completeness. Make necessary revisions to improve the flow and accuracy of the information.			

4. Gantt Chart You have to use the provided Gantt chart template.

Use the provided Gantt chart template to list all items from the Activity Definition along with relevant estimates and scheduling. Ensure that the Gantt chart reflects the activity definitions from Section 3. Track actual start times and durations. Besides including Gantt chart here, you should also submit your Gantt chart file separately.



Activity No. and N	Plan Start	Plan Duration	ual Start	al Duration	t Complete
1.Intiating Phase	1	30	1	0	0%
1.1 Define Project objective	1	5	1	5	0%
1.2 Identify Project Stakeholders	7	14	7	14	0%
1.3 Define Project Scope	19	5	19	5	0%
1.4 Weekly Project Meetings	4	1	4	1	0%
2. Planning Phase	19	4	19	4	0%
2.1 Develop Work Breakdown Structure	23	5	23	5	0%
2.2 Estimate Activities and	28	5	28	5	0%
2.3 Create Gantt Chart	33	4	33	4	0%
3.Execution Phase					0%
3.1 Problem background	37	8	37	8	0%
3.2 user requirements	45	5	45	5	0%
3.3 Strutural design	50	10	50	10	0%
3,4 documentations	60	27	60	27	0%
3.5 deployment	87	15	87	15	0%
					0%
4,1 status report	102	30	102	30	0%
4.2 Update plans	132	15	132	15	0%
					0%
5.1 Final performace review	147	10	147	10	0%
5.2 Prepare Final Report	157	5	157	5	0%
6.1 Frontend					0%