# Project Information – Kirke Application:

1. **Project Overview**:
   * Could you provide a brief description of the project and its primary purpose?
   * Who are the main stakeholders, and what are their expectations?
2. **Business Goals**:
   * What specific business problems are we solving with this project?
   * Are there any measurable success criteria or KPIs for the project?
3. **Target Audience**:
   * Who will be the end users of the project (e.g., internal teams, external customers)?
   * Are there specific user personas we should consider during development?

**Technical Details**

1. **Architecture**:
   * Are there any predefined architectural guidelines or standards we need to follow?
   * Will this project integrate with existing systems, or is it being developed from scratch?
2. **Technology Stack**:
   * Are there specific technologies, frameworks, or tools you’d prefer we use?
   * Do we need to account for scalability, redundancy, or cloud deployment?
3. **Data Requirements**:
   * What kind of data will the project handle (e.g., structured, unstructured, sensitive)?
   * Are there any specific compliance or security guidelines we must adhere to?

**Project Execution**

1. **Initial Deliverables**:
   * What is the first thing we are expected to deliver, and by when?
   * Is there an MVP (Minimum Viable Product) we should focus on initially?
2. **Team Roles and Responsibilities**:
   * Who will be responsible for various components of the project?
   * Are there other team members or external collaborators we should coordinate with?
3. **Challenges and Risks**:
   * Are there any known risks, challenges, or dependencies we should prepare for?
   * How frequently should we check in to update on progress?

**Communication and Documentation**

1. **Preferred Communication Channels**:
   * How should we communicate updates (e.g., emails, Slack, weekly meetings)?
   * Are there specific tools you’d like us to use for task tracking (e.g., Jira, Trello)?
2. **Documentation**:
   * Are there existing documents or resources we can review to understand the project better?
   * How detailed should our documentation be for deliverables?

**1. Programming Language and Framework**

**Question**:

* I understand the primary language is Python, and the framework is FastAPI. Are there any additional libraries or tools you’d like us to integrate (e.g., Pydantic, Uvicorn)?

**My Understanding**:

* The application will be built using Python and FastAPI.
* We will likely need libraries like Pydantic for data validation and Uvicorn for running the FastAPI server.
* I assume there might also be specific tools for logging, testing, or security (e.g., pytest, JWT). It would help to confirm if there are preferred libraries or tools you’d like us to use.

**2. Database**

**Question**:

* The project uses MongoDB, but could you confirm if it will be **on-premises** or **cloud-based**?
* If **on-premises**, do we have access to a testing database, or will we need to set one up?
* What are the credentials for accessing the database, and are there any specific roles or permissions we need to configure?

**My Understanding**:

* MongoDB will serve as the database.
* Currently, I am unsure whether this will be a cloud-hosted MongoDB (e.g., Atlas) or an on-premises database.
* If it is on-premises, I assume we might need access to a testing environment to avoid working directly with production data.
* Credentials (username, password) and role-based access will be essential for configuring secure database interactions.

**3. Deployment**

**Question**:

* Where will the application be deployed? (e.g., Docker, Kubernetes, cloud provider, or on-premises server?)
* If cloud-based, which cloud platform are we using (e.g., AWS, Azure, GCP)?
* If on-premises, could you provide the server details and credentials for deployment?
* Should we configure CI/CD pipelines for automated deployment? If yes, are there specific tools or platforms (e.g., Jenkins, GitHub Actions) preferred for this?

**My Understanding**:

* Deployment will be a key factor in how we structure the application.
* I assume Docker will be used for containerizing the application, with potential orchestration via Kubernetes if scalability is needed.
* If deployed on the cloud, my guess would be AWS (EC2 or ECS). If on-premises, we might need detailed server configurations.
* For CI/CD, I believe tools like Jenkins or GitHub Actions could help automate builds, testing, and deployment, but I’m unsure if there’s a preferred setup in place.

**4. API Context**

**Question**:

* Since the application involves an API called Kirke, could you clarify our role:
  + Are we acting as customers accessing Kirke to use its services?
  + Or are we building an intermediate application to process requests for others accessing Kirke?
* Is there any documentation available for the Kirke API that we should review?

**My Understanding**:

* My current understanding is that Kirke API plays a central role in the project.
* I’m unclear whether we’re building this application as a client (to consume Kirke services) or as a middleware layer for processing requests from others who will use Kirke.
* Documentation for the Kirke API would be critical in either case to ensure proper integration and implementation.

**5. Target Use Case**

**Question**:

* What is the primary goal of the application? For example:
  + Are we primarily exposing endpoints to external clients?
  + Is the focus more on data processing, aggregation, or transformation?

**My Understanding**:

* My assumption is that this project focuses on creating backend endpoints that interact with Kirke API and possibly MongoDB.
* The application might be handling data aggregation, transformation, or simple passthrough functionality, but further clarity is needed.
* If there’s a broader business goal (e.g., reporting, analytics), it would be helpful to understand that context.

**6. No UI**

**Question**:

* Since the project doesn't involve a UI, will the entire interaction happen via APIs?
* Should we account for any future UI components, or is this strictly a backend service?

**My Understanding**:

* My understanding is that this is a backend-only project, and all interactions will happen via API calls.
* If there’s a chance of a UI being added later, we might need to structure the backend in a way that supports it seamlessly.

**7. Initial Deliverables**

**Question**:

* What is the first deliverable we need to focus on? For example:
  + Setting up database connectivity?
  + Exposing an API endpoint?
* Are there any deadlines or milestones we should keep in mind?

**My Understanding**:

* My assumption is that setting up database connectivity and basic endpoints (e.g., a health check endpoint) would be the first priority.
* If there’s a Minimum Viable Product (MVP) goal or timeline, it would help to know the exact expectations for deliverables.

**8. Documentation**

**Question**:

* Should we start documenting the application design and architecture?
* Are there specific templates or formats you’d prefer for documentation?

**My Understanding**:

* My assumption is that initial documentation might include:
  + An architecture diagram showing how the application interacts with Kirke API and MongoDB.
  + API documentation for each endpoint.
* If there are existing templates or formats you’d prefer, it would help align our documentation efforts.

**9. Team Collaboration**

**Question**:

* Are there other team members or stakeholders involved who we should coordinate with?
* How frequently would you like updates on progress (e.g., daily stand-ups, weekly syncs)?

**My Understanding**:

* I assume I will be working closely with other developers, but I am unclear on the complete team structure.
* I expect regular updates will happen, but I’d like to confirm the frequency and preferred mode of communication.