# **🤖 LLM Task Execution Protocol**

This protocol outlines the rules, dependencies, and execution flow for any AI agent (GPT, Claude, or automated tool) tasked with generating code, content, or documentation for the Defeah project.

## **📌 1. Purpose**

The purpose of this protocol is to ensure that every task executed by an AI assistant:

* Respects project context and spiritual integrity
* Aligns with current development roadmap and product vision
* Produces technically and theologically sound output
* Avoids rework, misalignment, or unsafe content generation

This document acts as the **bootstrap context file** for LLM agents and is required for all automation workflows.

## **📚 2. Required Pre-Reading**

| **Task Type** | **Required Documents** |
| --- | --- |
| ✨ Feature Dev | Product Roadmap, PRD, Sprint Plan, Sprint Tasks, Technical Architecture |
| 🔧 Infra / DevOps | Dev Onboarding, CI/CD Config, Environment Config, Git Workflow Guide |
| 🐞 Bug Fix | Bug Report Template, Error Logging Plan, Sprint Tasks, Dev Docs |
| 📖 Theology Content | Theological Accuracy Guidelines, Prompt Curation Rules, Scripture Source List |
| 🧪 Test Writing | Sprint Tasks, Architecture, Environment Config, Feedback Logs |
| 📄 Documentation | All above, based on context of target document |

✅ Before beginning any task, the LLM must parse the above documents based on the task category.

## **🧠 3. Context Application Rules**

1. 📘 **Always consult the PRD and Roadmap** before generating new features or endpoints
2. 🙏 **Cross-check all spiritual/theological content** with the Theological Accuracy Guidelines
3. 🔐 **Use only referenced environment variables** from the Env Config Doc—never hardcode API keys
4. 🧩 **Reference Sprint Plans + Sprint Tasks** before starting any frontend/backend task
5. 🔄 **Adhere to Git Workflow**: Branch names, commit format, PR rules must be followed
6. 📎 **Traceability**: Each generated output must refer back to the sprint task or feature ID it supports

## **🛠️ 4. Task Execution Steps**

For every execution, follow this protocol exactly:

1. **Ingest task description** (provided by PM, dev, or automation agent)
2. **Pull relevant sprint plan and PRD sections** for this task
3. **Map task to domain** → Load all matching reference documents (from Section 2)
4. **Draft a short implementation plan** (architecture or strategy)
5. **Generate code / documentation / prompt** only after steps 1–4 are complete
6. **Validate output** against checklist:  
   * ✅ Matches PRD requirements
   * ✅ Theologically accurate (if spiritual)
   * ✅ References correct environment variables
   * ✅ Consistent with sprint boundaries and Git practices
7. **Attach references** to relevant docs in the final output comment or PR description

## **✅ 5. Output Guidelines**

* Output must be **formatted cleanly** (Markdown or code blocks)
* **Function/class/file names** must follow existing naming conventions (refer Git Workflow Guide)
* **Use env vars** as defined in Environment Config (dotenv in Flutter, secrets in Supabase/Firebase)
* **Document assumptions and link references** to PRD, sprint task ID, or theology rules
* **No inline secrets or credentials**
* **Include TODOs** if future considerations or developer handoff needed

Example:

final openAiKey = dotenv.env['OPENAI\_API\_KEY']; // Loaded from .env.dev

## **🚫 6. Red Flags & Failure Triggers**

AI output must be rejected and regenerated if any of the following are observed:

❌ **No link to PRD, roadmap, or sprint task** for the generated content  
 ❌ **Spiritual content lacks theological validation** via accuracy checklist  
 ❌ **Hardcoded credentials** or API endpoints not managed via .env  
 ❌ **Incorrect Git branch naming or PR format** ❌ **Output not aligned with environment (dev/staging/prod)** ❌ **Use of Bible sources not listed in Scripture Source List**

## **📌 Final Notes**

This protocol ensures that AI remains a trusted co-developer on the Defeah Bible Study App. It must be applied uniformly across all LLM-based task runners, whether interactive or autonomous.

File: LLM\_EXECUTION\_PROTOCOL.md  
 Owner: Project Architect / DevOps Lead