**TARDIS AI Chatbot challenge**

**Problem Statement:**

TARDIS is a very big project consisting of multiple various components providing different functionality. Its documentation is enormous and may be hard to navigate, especially for new users.

Customer support takes a lot of developers’ time. Most of the questions can be answered by reading the documentation.

**Requirements and Expectations:**

TARDIS ChatBot:

* Has an easy-to-use user interface
* Answers documentation-based questions
* Guides the user with instructions on integration with TARDIS components (exposing/subscribing to an API, using Horizon events, CloudWalker file transfer, etc.)
* Helps to solve the problem based on the user’s input (when something is not working on the user’s side)
* Provides different levels of information details (e.g. for a business analyst – high-level description, for a developer – technical details)
* Receives feedback on its operation and quality of answers
* Improves (learns) from the feedback

Additionally, it:

* Recognizes a problematic component based on logs, provided by the user
* Provides answer with images (there are a lot in documentation)
* Is embedded in Webex/Microsoft Teams
* If request/issue cannot be resolved by the bot, it: either sends a notification for human help or creates a Jira ticket
* Creates tickets in Jira:
  + Fills it in with all necessary data (requirements are provided in a separate file)
  + Automatically assigns tickets to the responsible team
  + Accepts attachments such as screenshots, logfiles, etc.
  + Sets user (defined by email) as the Reporter
  + Provides the user with the link to the created ticket
* Acts like a bot in Jira: analyzes the tickets and replies to them before human interaction
* Has the possibility for admins to upload new documents and update old ones, to add short-term notifications (e.g. “*there is an outage on prod, our services may work with disruptions, we are working on it and will notify you when the problem is resolved*”); corrects its answers based on the new information
* Derives documents from a git repository (GitLab)
* Provides the possibility for admins to correct its answers (teaching it to answer correctly)
* Collects statistics on FAQ
* 🌟 Your ideas are welcome!

**Evaluation criteria:**

* Accuracy and relevance of the answers
* Efficiency
* Depth
* Number of features implemented