Submitted To,

Amit Kumar Mondal

Associate Professor

Computer Science & Engineering Discipline

Khulna University, Khulna

Submitted By,

Sharmika Das Banhi

Student ID: 210204

Computer Science & Engineering Discipline

Khulna University, Khulna

An Assignment on Extract various Artifacts (such as source code, commit messages and so on) of the following open-source software systems:

1. Azure Java SDK
2. ChatGPT

Azure Java SDK:

Artifact	Description	Purpose	Creation Method
Codebase	The collection of source code files forming the software system.	Foundation of the software containing all instructions and logic.	Written and contributed to by programmers using development tools.
Core Executable	The main executable file(s) constituting the functional software application.	The user interacts with this executable form of the software.	Created through the compilation of the source code using a build process.
Commit	Represents a specific version of the codebase, including changes made by developers.	Keeps track of changes, allowing collaboration and version control.	Developers make commits using version control tools (e.g., Git), describing changes made.
Workflow	A series of automated steps defining the process of building, testing, and deploying software.	Ensures consistency and efficiency in the development lifecycle.	Defined in workflow configuration files (e.g., GitHub Actions YAML files).
Unit of Work	A set of related tasks or activities needed to achieve a specific goal.	Manages and tracks progress on a feature or bug fix.	Tracked in project management tools or issue trackers, associated with specific issues or tasks.
Readme	A document providing information about the software, its usage, and other details.	Serves as documentation for users and developers.	Created and maintained by developers, often written in Markdown format.
Activity	Records of user interactions and system events within the software project.	Provides insights into how users engage with the system.	Generated automatically by tracking user actions or events in the application.
Action	Automated tasks or processes triggered by specific events in the development lifecycle.	Streamlines repetitive tasks such as building, testing, and deploying code.	Defined in configuration files (e.g., GitHub Actions YAML files) and associated with specific events.

Releases	Specific versions of the software considered stable and ready for deployment.	Indicates significant milestones or improvements in the software.	Tagged in the version control system and associated with specific commit states.
Documentation	Comprehensive documentation explaining how to use the Azure Java SDK.	Guides users and developers on SDK usage.	Authored and maintained by developers, often generated from comments in the source code and supplemented with additional information.
Security Policies	Guidelines and policies related to security practices and considerations for the Azure Java SDK.	Ensures secure coding practices and user data protection.	Developed and maintained by security experts and integrated into the overall development process.

ChatGPT:

Artifact	Description	Purpose	Creation Method
Codebase	The collection of source code files forming the ChatGPT system.	Foundation of the ChatGPT model or API containing all instructions and logic.	Developed and maintained by researchers and engineers using specialized development tools.
Core Executable	The main executable file(s) constituting the functional ChatGPT model or API.	Users interact with this executable form for generating human-like text.	Created through the compilation and packaging of the model or API code using a build process.
Commit	Represents a specific version of the ChatGPT model or API, including changes made by developers.	Keeps track of changes, allowing collaboration and version control.	Developers make commits during the training process and model or API updates, using version control tools (e.g., Git).
Workflow	A series of automated steps defining the process of training, testing, and deploying ChatGPT.	Ensures consistency and efficiency in the development lifecycle.	Defined in workflow configuration files, specifying steps in the training and deployment pipelines (e.g., GitHub Actions YAML files).
Unit of Work	A set of related tasks or activities needed to achieve a specific goal in ChatGPT development.	Manages and tracks progress on improving the model's performance or adding features.	Tracked in project management tools or issue trackers, associated with specific research or development tasks.
Readme	A document providing information about the ChatGPT model or API, its usage, and other details.	Serves as documentation for users, developers, and researchers.	Created and maintained by researchers and developers, often written in Markdown format.
Activity	Records of user interactions and system events within the ChatGPT model or API.	Provides insights into how users interact with the generated text.	Generated automatically by tracking user interactions or events during API usage.

Action	Automated tasks or processes triggered by specific events in ChatGPT development.	Streamlines repetitive tasks such as model training, testing, and deployment.	Defined in configuration files (e.g., GitHub Actions YAML files) and associated with specific events.
Releases	Specific versions of the ChatGPT model or API considered stable and ready for deployment.	Indicates significant milestones or improvements in the model or API.	Tagged in the version control system and associated with specific commit states during official releases.
Documentation	Comprehensive documentation explaining how to interact with the ChatGPT model or API.	Guides users, developers, and researchers on using and integrating ChatGPT effectively.	Authored and maintained by researchers and developers, often generated from model documentation and supplemented with additional information.
Security Policies	Guidelines and policies related to security practices and considerations for ChatGPT.	Ensures secure usage and handling of sensitive information by incorporating best security practices.	Developed and maintained by security experts, researchers, and developers, integrated into the overall development and deployment processes.