**An Assignment on** "Extract various Artifacts (such as source code, commit messages and so on) of the following open source software systems: Azure Java SDK, ChatGpt.

**Course Name: Software Development Lab**

**Course No.: CSE 3106**

<u>**Submitted by:**</u>

Parthoraj Nandi

Student ID: 210202

Computer Science & Engineering Discipline,

Khulna University, Khulna.

<u>Submitted to:</u>

Dr. Amit Kumar Mondal

Associate Professor

Computer Science & Engineering Discipline

Khulna University, Khulna

**Azure Java SDK:** The Azure SDK for Java allows Java developers to build applications that leverage various Azure services, such as storage, compute, networking, databases, and more.

| Artifact Name | What is it | Why it is used | How is it created |
|---|---|---|---|
| doc | Documentation for the current repository | To store advanced documentation for developers of SDK. | Often created manually in markup languages or tools to generate documentation from inline comments in the source code. |
| eng | A group of engineering tools or utilities that the build configuration uses. | Used to carry out routine or repeated tasks. | Scripting languages are used by programmers to write it. |
| samples | A compilation of instructions, sample code, and application examples. | To aid in the learning and modification of developers. | From sample data sets, example setups, or short bits of code. |
| Azure management client library | A higher-level, object oriented API | A method of managing Azure resources that puts an emphasis on uniformity, conciseness, and simplicity. | Through an already-existing system for user and developer input. |

| Comments | Annotations within the code explaining its functionality | Used for documentation and to enhance code readability. | Programmers write comments within the code to explain complex logic or functionality. |
|---|---|---|---|
| Core Executable | The main executable of the software | The compiled version of the software that can be executed to run the application. | Created through the compilation process, converting source code into machine-readable instructions. |
| Actions | Automated workflows triggered by events | Used for automating tasks like testing, building, and deployment based on specific events. | Configured through YAML files defining workflows and actions to be performed. |
| Codebase | Implemented software code | The fundamental collection of source code files that constitute the software application. | Programmers and tools write or create it. |
| Core executables | Executables with a user-centered design. | Combining source code and graphical user interface (GUI). | Compiled from the source code. |
| Pull Requests | Proposed changes submitted by a developer | Allows developers to review, discuss, and merge code changes into the codebase. | Developers create a branch, make changes, and submit a pull request for review. |

| | | | |
|---|---|---|---|
| Discussions | Conversations around code or project-related topics | Facilitates communication amongteam members regarding code implementation or project decisions. | Users initiate discussions, and team members participate in conversations. |
| Wiki | Collaborative documentation space | A place to store and share project-related documentation and information. | Created and edited by project contributors, usually using a markup language. |
| Security | Information related to security measures and vulnerabilities | Monitors and addresses security concerns within the codebase. | Security information is managed and updated by the development team, addressing reported vulnerabilities. |
| Readme | Project introduction and setup instructions | Provides essential information about the project, helping users and contributors understand its purpose and how to use it. | Created and maintained by project contributors in Markdown or another markup language. |
| Activity | Record of project-related events and changes | Offers insights into the project's recent activities, such as code changes, pull requests, and discussions. | Automatically generated by the platform based on user actions and project events. |
| Customer Property | Custom information related to user preferences or settings | Captures personalized data that enhances the user experience or tailors the software to individual needs. | Created and managed based on user inputs and preferences, often stored in configuration files or databases. |

| Releases | Versioned releases of the software | Represents stable and tested versions of the software for distribution. | Created by the development team when they decide a set of features is ready for a stable release. |
| --- | --- | --- | --- |

**ChatGPT:** ChatGPT is a language model developed by OpenAI, based on the GPT (Generative Pre-trained Transformer) architecture.

| Artifact Name | What is it | Why it is used | How is it created |
| --- | --- | --- | --- |
| | | | |
| assets | Usually refers to binary files or other non-code files, such as documents, images, configuration files, or other binary files. | Different uses, such as using pictures for UI components or logos, documents for documentation, and configuration files for setup. | Tools for graphic design are used for images, text editors are used for documents, and configuration files are created menu-driven. |
| Pull Requests | Proposed changes submitted by a developer | Allows developers to review, discuss, and merge code changes into the ChatGPT codebase. | Developers create a branch, make changes, and submit a pull request for review. |

| Actions | Automated workflows triggered by events | Used for automating tasks like testing, building, and deployment based on specific events in the ChatGPT repository. | Configured through YAML files defining workflows and actions to be performed. |
|---|---|---|---|
| Discussions | Conversations around code or project-related topics | Facilitates communication among team members regarding ChatGPT's code implementation or project decisions. | Users initiate discussions, and team members participate in conversations. |
| script | Code files created with scripting languages such as JavaScript, Python, or Shell. | To carry out operations, automate tasks, or carry out tasks. | Integrated development environments (IDEs) are used by developers to create, modify, and save these files. |
| src | Usually refers to a project's source code files and stands for "source." | Comprises the primary code that carries out the project's features and serves to specify the logic and functioning of the software. | Text editors or IDEs are used by developers to construct source code files. |

| public | Any files intended for usage by the whole public. Open-source licensing, documentation, and other things might be involved. | Information exchange, usage rules, and licensing specifics that promote developer cooperation. | Typically, documents are either manually made or written in markdown formats. |
|---|---|---|---|
| Codebase | Implemented software code | The fundamental collection of source code files that constitute the ChatGPT application. | Programmers and tools write or create it. |
| Core Executable | The main executable of the software | The compiled version of ChatGPT that can be executed to run the application. | Created through the compilation process, converting source code into machine-readable instructions. |
| Comments | Annotations within the code explaining its functionality | Used for documentation and to enhance code readability in ChatGPT. | Programmers write comments within the code to explain complex logic or functionality. |

| Wiki | Collaborative documentation space | A place to store and share ChatGPT-related documentation and information. | Created and edited by project contributors, usually using a markup language. |
|------|-----------------------------------|--------------------------------------------------------------------------|------------------------------------------------------------------------------|
| Security | Information related to security measures and vulnerabilities | Monitors and addresses security concerns within the ChatGPT codebase. | Security information is managed and updated by the development team, addressing reported vulnerabilities. |
| Readme | Project introduction and setup instructions | Provides essential information about ChatGPT, helping users and contributors understand its purpose and how to use it. | Created and maintained by project contributors in Markdown or another markup language. |
| Activity | Record of project-related events and changes | Offers insights into recent activities in the ChatGPT repository, such as code changes, pull requests, and discussions. | Automatically generated by the platform based on user actions and project events. |

| Customer Property | Custom information related to user preferences or settings in ChatGPT | Captures personalized data that enhances the user experience or tailors ChatGPT to individual needs. | Created and managed based on user inputs and preferences, often stored in configuration files or databases. |