# Knowledge-Based Assessment (KBA)

**Assessment Resources:**

|  |
| --- |
| Marking key available for lecturer via Blackboard.  Students may refer to the lecture material (GitHub and Blackboard) in formulating their answers. |

**Assessment Instructions:**

|  |
| --- |
| Students must attempt all questions. Answer succinctly using full sentences. At most three paragraphs are expected per answer.  All answers must be at the student’s own words – copying generated code or answers from ChatGPT or other AI tools is **strictly** prohibited.  Please ensure that all instructions are followed **carefully**, and submissions are well-organized, clearly labelled, and accompanied by any necessary explanations or justifications. |

# Select and Contribute to Open-Source Projects

1. **Identify an open-source target program requiring maintenance.**

* **Task**: Explore the open-source landscape and choose a Python-based project that you believe you can contribute to.
* **Instructions**: Provide the name and a link to the chosen project/program. Include a short justification explaining why you chose this program and how you believe you can contribute (broadly).

Not visual desktop access (NVDA)

https://www.nvaccess.org/download/

https://github.com/nvaccess/nvda

I hve worked with a person who used screen readers because they were blind and I saw how important it is. Also my own eyesight is declining.

1. **Establish and confirm the target program's requirements and specifications.**

* **Task**: Focus on understanding the current state of the project and areas where there is a need for improvements.
* **Instructions**: Outline where you believe the core development focus is and why.

NVDA is basically feature complete. They are asking for feature suggestions, Translations and bug fixes.

From the git hub branches and document the main focus is keeping updated with compatibility when interacting with other applications that might crash it, because applications are also updating . new versions of programming languages, like new version of python, and new versions of windows. From the <https://www.nvaccess.org/get-help/> the is also a emphases on translations to other spoken languages so they can reach more with they servce.

1. **Examine online resources applicable to the target program.**

* **Task**: Identify specific resources that are relevant to the chosen project, such as official documentation, forums, or community chat platforms.
* **Instructions**: Provide a list of these resources with brief descriptions of their relevance and how they can aid in your contribution process.

<https://www.nvaccess.org/files/nvda/documentation/developerGuide.html> A comprehensive to how the community develops NVDA and how it can be contributed to.

<https://github.com/nvdaaddons/DevGuide/wiki/NVDA-Add-on-Development-Guide> A more focused guide to make add-ons.

<https://github.com/nvaccess/nvda/tree/master/projectDocs> The root for the project document.

<https://github.com/nvaccess/nvda/blob/master/.github/CONTRIBUTING.md> A focused documentation on contributing.

<https://github.com/nvaccess/nvda/blob/master/projectDocs/community/readme.md> A community readme that has all the important communication places, important documents and guides listed.

[https://app.gitter.im/#/room/](https://app.gitter.im/" \l "/room/)#nvaccess\_NVDA:gitter.im Developer chat

<https://groups.io/g/nvda-devel>Developer email groups

1. **Set up the target project locally.**

* **Task**: Instead of downloading binaries, clone the project repository and set it up on your local machine.
* **Instructions**: Describe the process you followed to set up the project, including any dependencies you installed. Highlight any challenges faced and how you overcame them. If you used instructions from the project's documentation, provide a link to the relevant section. If you did not, explain why you believe the project didn't have instructions for contributing.

1. **Access and analyse the supporting documentation of the target program.**

* **Task**: Dive deep into the project's documentation, focusing on setup instructions, contribution guidelines, and code of conduct.
* **Instructions**: Provide insights and key takeaways from the documentation. Mention any gaps or areas of improvement you identified.

This is Australia analyse is not spelt with a z.

1. **Engage with the Open-Source Community.**

* **Task**: Join an online community or forum related to your chosen program. Initially, focus on passive participation to understand the community dynamics.
* **Instructions**: Provide screenshots or other evidence of your engagement. Reflect on your experience joining the community and any insights gained from passive participation.

1. **Run the project locally.**

* **Task**: Execute the project on your local machine, ensuring it runs as expected.
* **Instructions**: Detail the steps you took to run the project and any challenges faced. If there were deviations from the documentation or additional steps required, document those as well.

1. **Analyze the project's issue database.**

* **Task**: Explore the project's issues or bug database. Prioritize bugs or tasks labelled as "good first issues" or "beginner-friendly."
* **Instructions**: What system does the project use to track issues? What are some labels you identified that are relevant. Provide screenshots of the bugs and tasks you've identified. Describe your criteria for selection and any filters or tags you used to prioritize them.

1. **Familiarize yourself with the project's development tools and environment.**

* **Task**: Understand and set up any tools or environments essential for the project, such as virtual environments, linters, or testing frameworks.
* **Instructions**: Detail the tools and environments you used during this process. Highlight any challenges faced and how you addressed them. What CI/CD tools does the project use? What are the project's testing requirements? What are the project's linting requirements?