## **Requirement Analysis: In-built IDE for Basic Coding Activities**

This document outlines the requirements for a built-in integrated development environment (IDE) focused on supporting basic coding activities within the integrated learning environment (ILE) platform.

****1. Functional Requirements:****

****1.1 Supported Languages:****

The IDE should support at least one popular and beginner-friendly coding language, such as:

****Python****

****JavaScript****

****Java**** (Consider complexity for beginners)

****C++**** (Consider complexity for beginners)

****1.2 Coding Features:****

Provide basic functionalities for coding, including:

****Code editing:**** Allow users to write, edit, and save code files.

****Syntax highlighting:**** Highlight different code elements (keywords, functions, variables) for improved readability.

****Basic code completion:**** Suggest code snippets or function definitions based on user input (optional).

****Indentation and formatting:**** Automatically format code for consistency and improved readability.

****Basic debugging tools:**** Provide functionalities like setting breakpoints and inspecting variables to help identify and fix errors (optional).

****1.3 User Interface:****

Design a ****user-friendly interface**** suitable for beginners:

****Clear layout:**** Organize code editing space, buttons, and other elements logically for intuitive use.

****Minimal distractions:**** Prioritize essential features and minimize unnecessary visual elements to avoid overwhelming users.

****Contextual help:**** Offer relevant help documentation or tutorials within the IDE (optional).

****1.4 Integration with Other Platform Features:****

****Course materials:**** Allow students to directly open and edit code practice files assigned within their courses.

****Collaboration:**** Enable students to collaborate on code projects within the IDE (consider real-time collaboration features if applicable).

****Submission:**** Facilitate easy submission of completed assignments directly from the IDE to the platform.

****2. Non-Functional Requirements:****

****Performance:****

The IDE should operate smoothly and responsively to user actions and code editing.

****Security:****

Implement security measures to prevent code injection vulnerabilities or unauthorized access to user projects.

****Accessibility:****

Consider accessibility features for users with disabilities, such as keyboard shortcuts and screen reader compatibility.

****3. Success Criteria:****

Users can effectively write, edit, and debug basic code within the IDE.

The IDE supports selected programming languages and provides necessary features for beginners.

The user interface is clear, intuitive, and minimizes distractions.

The IDE integrates seamlessly with other platform functionalities (course materials, collaboration, submission).

****4. Open Questions:****

What level of complexity should be offered in the code completion or debugging features?

Should the IDE support additional features like code version control or project management (optional)?

How will the IDE handle potential security risks associated with user-submitted code?