GitHubGitHub is a web-based version control platform for in collaborative software development utilizing Git. Developers can use it to host and manage code repositories, as well as enable collaboration   
Main Offers/of Imobie Primary Functions/Features  
Repositories: Containers for project code (repo = repository) with features provided by version control systems.  
Branches: Independent lines of development within a repository.  
Pull Requests: Suggestions on how your code changes could be integrated back into their original branch or another branch.  
Problems: Bug/Feature tracking and management.  
Actions: CI/CD pipelines that can be automated within your workflows.  
Team: Methods for code reviewing, tracking issues and projects & discussion.  
Repositories on GitHub  
What is a GitHub Repository?  
A GitHub repository is The place where your project files are saved with their revision history is called a GitHub Repository. It can be public or private.

Creating a New Repository

* Go to GitHub and log in.
* Create a new repository by clicking on the + icon in upper right corner and click New Repository.
* Enter the repository name, show a brief description (optional) and set visibility to public or private
* Initialize with a README,. / /gitignore/contentassistUnlike license or gitignore if you prefer.
* Click "Create repository."
* Key Components of a Repo

1. README. md: Description of what the project is
2. LICENSE - Describes the type of license used for that code.
3. gitignore: This file contains a list of files and directories which do not need to be tracked in git.
4. src/: This will contain all the code that we write.
5. tests/: Directory which contains the test codes.
6. docs/: Project documentation
7. Version Control with Git

Concept of Version Control

Version control is the management of changes to texts, program codes or other collections of information. It permits many developers to function on a project at the same time; keeps a history of changes and can bring back previous versions.

GitHub and Version Control

Github makes version control more better, by providing:

Central host: A place where repositories can be accessed to collaborate.

Pull requests: Optimize code review and feedback.

Commits: Get a visual history of changes, and branches.

Integration - Smooth Integrations

Pull Requests & Code Review

What is a Pull Request?

Pull request(PR) means requesting changes from one branch to the other. Enables other team members to look at code and have discussions.

Create and Review a Pull Request

Push your branch to GitHub.

Browse the repo on GitHub.

New pull request->Select the branch to merge

Add a title and description.

Click "Create pull request."

Reviewing a PR:

Navigate to the pull request.

Examine the changes, add comments if necessary

Approve or request changes.

Merge the pull request once it is approved.

GitHub Actions

What are GitHub Actions?

GitHub Actions are tools to do the automation of building, testing and deploying code. They are CI/CD pipelines right inside of GitHub.

Introduction to Visual Studio

What is Visual Studio?

Visual Studio is an application development tool from Microsoft. Support for different programming languages and utilities like debuggers, version control etc are available through this.

Key Features

Perform Code Editing: With Intellisense, advanced code editor.

Debugging: Built-in debugging and testing tools

Version control: In fact, this has in-built Git support.

Extensions: Additional tools and features marketplace.

Different from to Visual Studio Code

Visual Studio: Make sure to download the full-featured IDE Tools!

Visual Studio Code : Lightweight code editor which can be customized through plugins and serves good for all kind of development work.

How To Integrate Github With Visual Studio

How to add GitHub Repository - Steps

Then, open Visual Studio and navigate to Team Explorer.

Go to clone a repository and download it

supply the GitHub repository URL.

Choose a a local path and click "Clone."

Improving Development Workflow

Integration provides:

Hot code load: Pull, push and commit changes directly from the IDE

Integration with GitHub: Consolidated view of sources from multiple projects in issues, pull requests and more.

Increased productivity: coding rather than switching between different contexts.

Debugging in Visual Studio

Debugging Tools

Breakpoints - pausing the execution at certain lines of code.

Watch window: Observe the behavior of variables and expressions.

Call stack - List of functions and the sequence function calls

Immediate Window: Directly run the code and get expression value during debugging phase.

Identifying and Fixing Issues

Breakpoint in sections of code that you are suspicious about.

This will launch the application in debug mode.

Examine the value of variables and check your call stack.

Adapt the code based on discoverings, keep debugging until we solve this issues.

Collaborative Development with GitHub and Visual Studio

Real-World Example

Task: Create a web-platform.

GitHub: Host

GitHub: Repositories, Issue Tracking and Continuous Integration using Actions

Visual Studio: Write code, interact with GitHub and build tightly integrated features.

Benefits:

Versioning - Manually keeping track and handling changes.

Communication -- Pull Requests or The Code Review Way

AUTOMATION: CI/CD pipeline to maintain quality of code.

Source control: Keep tabs on modification.

Work together: Manage pull requests and code reviews effortlessly.

CI/CD pipelines (Automated): Ensure code quality with CI/CDAutomation.