Blockchain tutorial using C# implementation

Tech Week 2022 LaSalle College

Welcome to this tutorial, which you can use as a reference for the presentation at LaSalle College Montreal's Tech Week.

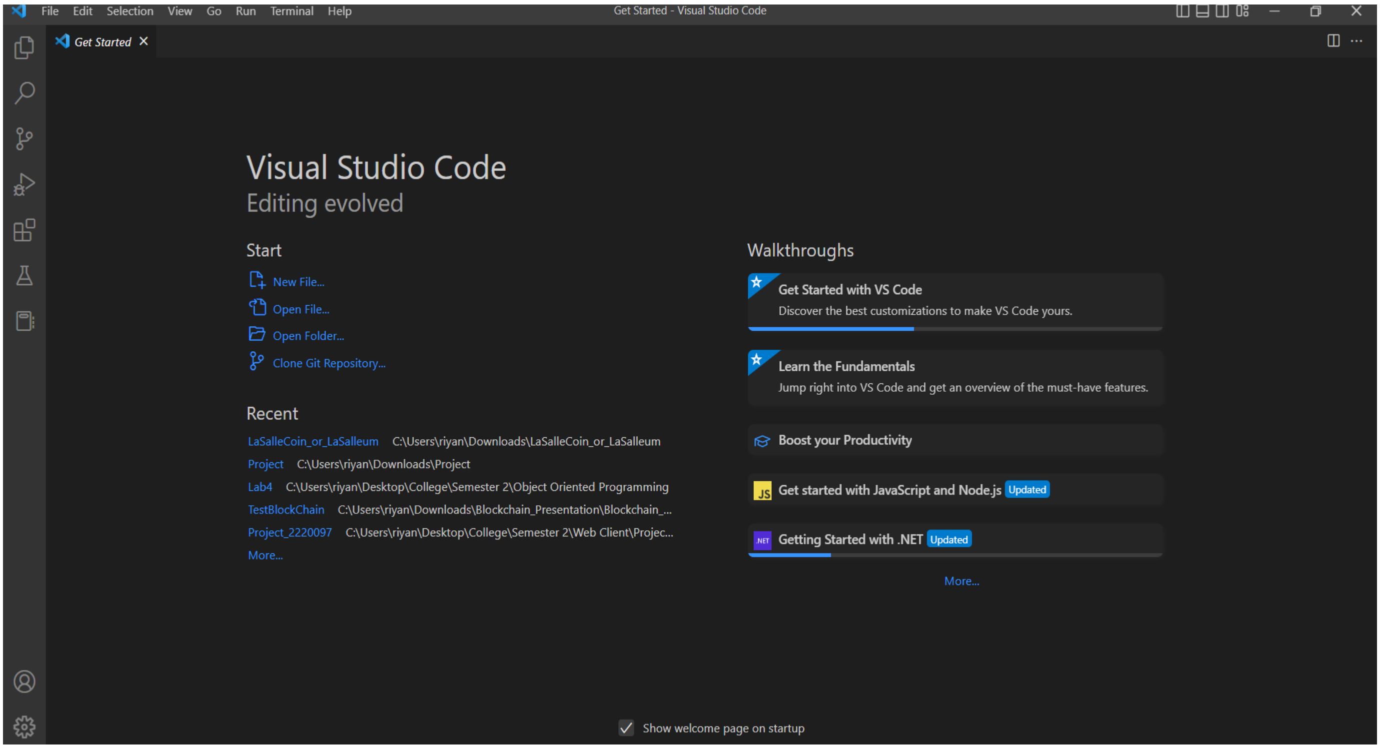
This information will help you get going.

Let's begin!

Visual Studio Code Installation

Every platform, including Linux, macOS, and windows, offers Visual Studio Code for free.

1. The version we'll be using is 1.73.1.
2. Visit the following website: https://code.visualstudio.com/download to begin the download.
3. Install it after the download is complete.
4. After the installation is finished, you should be able to access VS Code.



Discover the best customizations to make VS Code yours.

Jump right into VS Code and get an overview of the must-have features.

LaSalleCoin_or_LaSalleum C:\Users\riyan\Downloads\LaSalleCoin_or_LaSalleum
Project C:\Users\riyan\Downloads\Project
Lab4 C:\Users\riyan\Desktop\College\Semester 2\Object Oriented Programming
TestBlockChain C:\Users\riyan\Downloads\Blockchain_Presentation\Blockchain_...
Project_2220097 C:\Users\riyan\Desktop\College\Semester 2\Web Client\Projec...

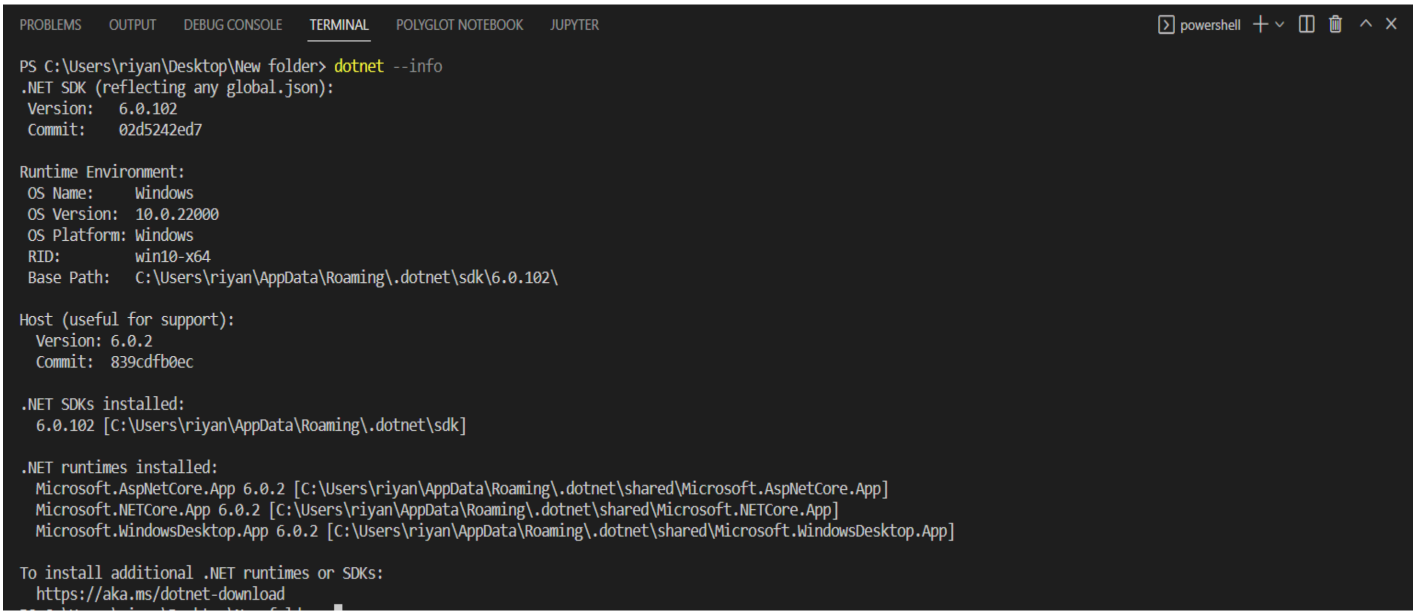
After successfully installing VS Code, we move forward with Installing DotNet.

.Net Installation

1. If you have successfully installed VS Code, you can now add .NET support by installing the .NET Extension Pack, which includes these extensions:
   - C# for Visual Studio Code
   - Ionide for F#
   - Jupyter Notebooks
   - .NET Interactive Notebooks

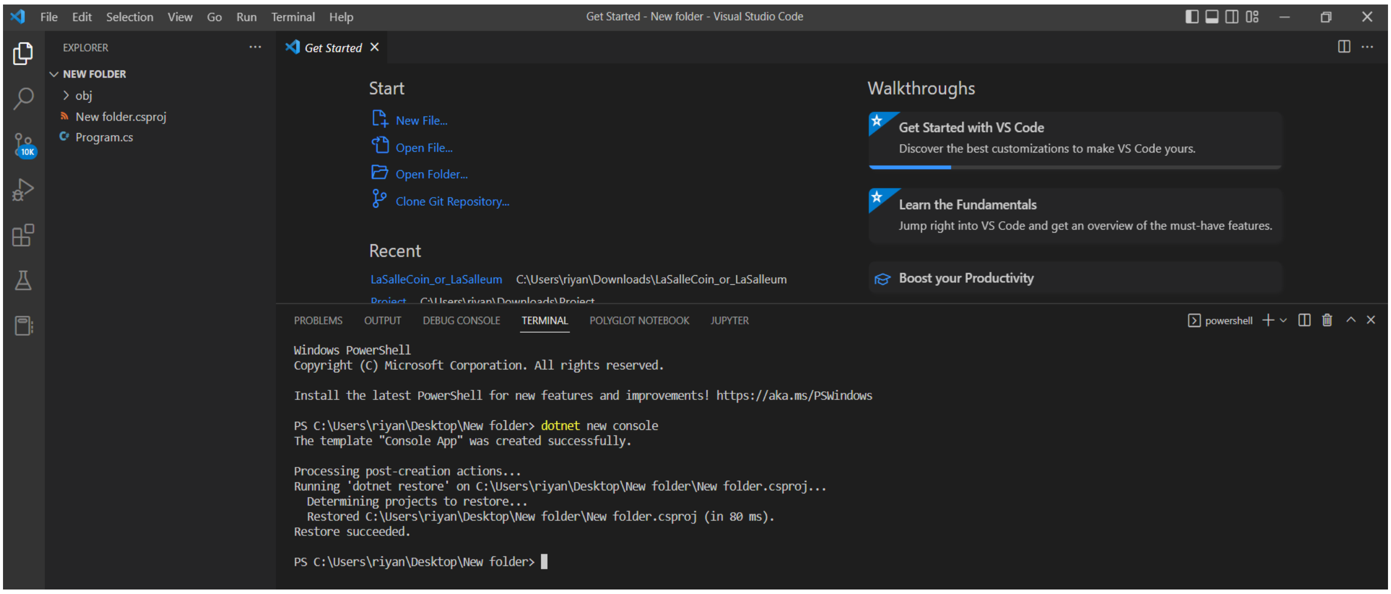
2. If the download of the extension is successfully completed, we move on to download the .NET SDK on our local environment. The .NET SDK is a software development environment used for developing .NET applications.

3. We can check the information/successful Implementation of dotnet by using the command dotnet --info as shown below.



```
PS C:\Users\riyan\Desktop\New folder> dotnet --info
.NET SDK (reflecting any global.json):
 Version:   6.0.102
 Commit:    02d5242ed7

Runtime Environment:
 OS Name:     Windows
 OS Version:  10.0.22000
 OS Platform: Windows
 RID:         win10-x64
 Base Path:   C:\Users\riyan\AppData\Roaming\.dotnet\sdk\6.0.102\

Host (useful for support):
  Version: 6.0.2
  Commit:  839cdfb0ec

.NET SDKs installed:
  6.0.102 [C:\Users\riyan\AppData\Roaming\.dotnet\sdk]

.NET runtimes installed:
  Microsoft.AspNetCore.App 6.0.2 [C:\Users\riyan\AppData\Roaming\.dotnet\shared\Microsoft.AspNetCore.App]
  Microsoft.NETCore.App 6.0.2 [C:\Users\riyan\AppData\Roaming\.dotnet\shared\Microsoft.NETCore.App]
  Microsoft.WindowsDesktop.App 6.0.2 [C:\Users\riyan\AppData\Roaming\.dotnet\shared\Microsoft.WindowsDesktop.App]

To install additional .NET runtimes or SDKs:
  https://aka.ms/dotnet-download
```

4. First, create a new folder and open it in Visual Studio Code. Then, run the command to create a new project using the dotnet framework: dotnet new console, as displayed below.



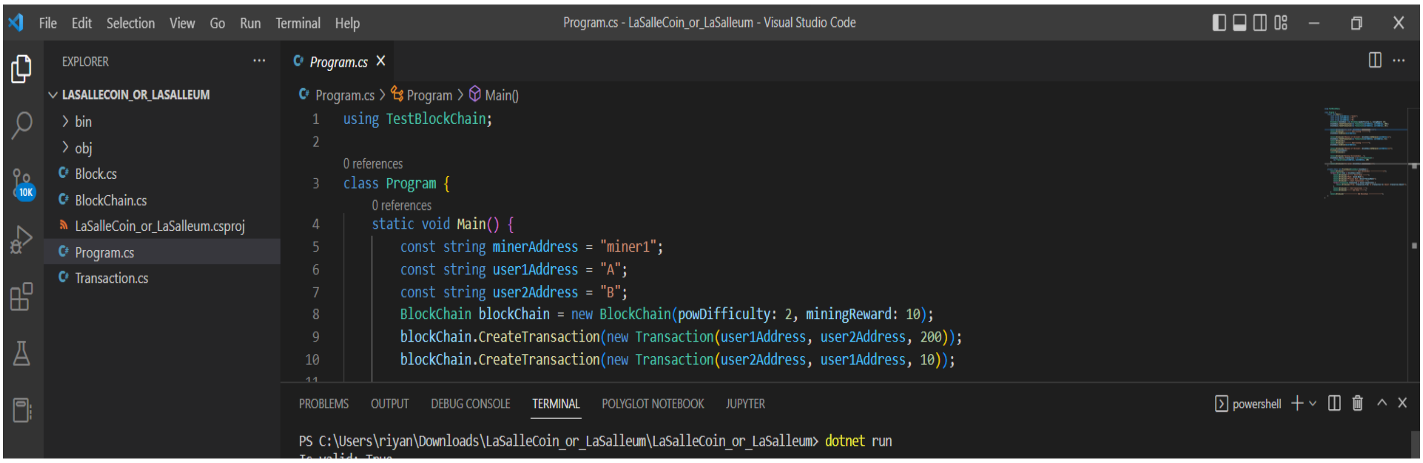
Discover the best customizations to make VS Code yours.

Jump right into VS Code and get an overview of the must-have features.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\riyan\Desktop\New folder> dotnet new console
The template "Console App" was created successfully.

Processing post-creation actions...
Running 'dotnet restore' on C:\Users\riyan\Desktop\New folder\New folder.csproj...
  Determining projects to restore...
  Restored C:\Users\riyan\Desktop\New folder\New folder.csproj (in 80 ms).
Restore succeeded.

PS C:\Users\riyan\Desktop\New folder>
```

5. We may now begin creating new files, such as the one designated as Program.cs, and begin putting our code for a fundamental explanation of Blockchain into practise.

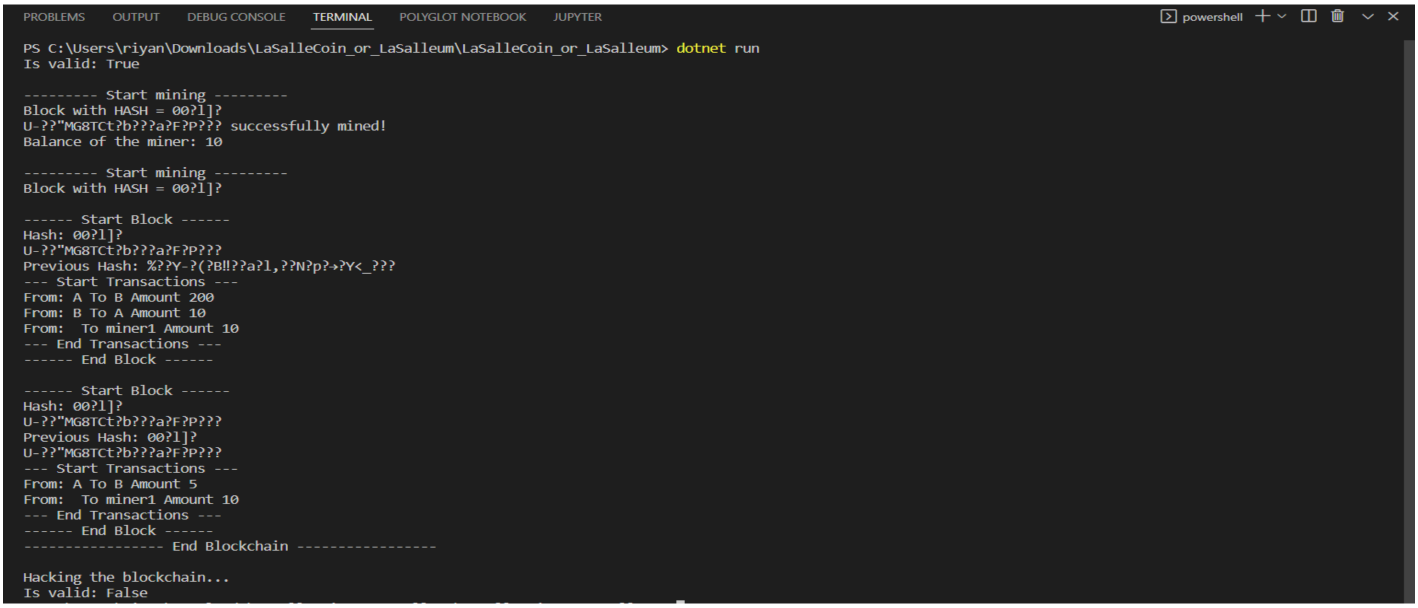
6. Once our code has been successfully implemented, we launch it using the command: dontnet run, as shown below.


File Edit Selection View Go Run Terminal Help
Program.cs - LaSalleCoin_or_LaSalleum - Visual Studio Code
EXPLORER
LASALLECOIN_OR_LASALLEUM
bin
obj
Block.cs
BlockChain.cs
LaSalleCoin_or_LaSalleum.csproj
Program.cs
Transaction.cs
Program.cs > Program > Main()
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL POLYGLOT NOTEBOOK JUPYTER
powershell

```csharp
using TestBlockChain;

0 references
class Program {
    0 references
    static void Main() {
        const string minerAddress = "miner1";
        const string user1Address = "A";
        const string user2Address = "B";
        BlockChain blockChain = new BlockChain(powDifficulty: 2, miningReward: 10);
        blockChain.CreateTransaction(new Transaction(user1Address, user2Address, 200));
        blockChain.CreateTransaction(new Transaction(user2Address, user1Address, 10));
```

```
PS C:\Users\riyan\Downloads\LaSalleCoin_or_LaSalleum\LaSalleCoin_or_LaSalleum> dotnet run
```

7. Our Output should look something like this :


PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL POLYGLOT NOTEBOOK JUPYTER
powershell

```
PS C:\Users\riyan\Downloads\LaSalleCoin_or_LaSalleum\LaSalleCoin_or_LaSalleum> dotnet run
Is valid: True

--------- Start mining ---------
Block with HASH = 00?l]?
U-??"MG8TCt?b???a?F?P??? successfully mined!
Balance of the miner: 10

--------- Start mining ---------
Block with HASH = 00?l]?

------ Start Block ------
Hash: 00?l]?
U-??"MG8TCt?b???a?F?P???
Previous Hash: %??Y-?(?B‼??a?l,??N?p?→?Y<_???
--- Start Transactions ---
From: A To B Amount 200
From: B To A Amount 10
From:  To miner1 Amount 10
--- End Transactions ---
------ End Block ------

------ Start Block ------
Hash: 00?l]?
U-??"MG8TCt?b???a?F?P???
Previous Hash: 00?l]?
U-??"MG8TCt?b???a?F?P???
--- Start Transactions ---
From: A To B Amount 5
From:  To miner1 Amount 10
--- End Transactions ---
------ End Block ------
---------------- End Blockchain ----------------

Hacking the blockchain...
Is valid: False
```