

BITS, Pilani – Pilani Campus

CS F452 (Blockchain Technology)

Laboratory 1 (Week-1)

AIM:

To know about the tools and their installation procedure required to build a Web3 application.

Tools Required:

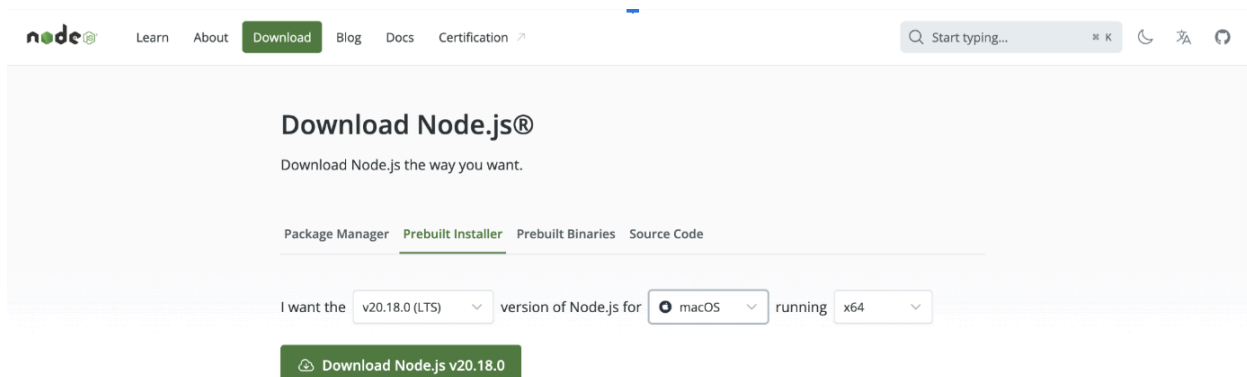
1. Node.js
2. Ganache
3. Truffle
4. Remix IDE
5. MetaMask

Node.js

Node.js allows you to **run JavaScript** on your local machine, which is critical for **managing the development environment**. It also includes a package manager, **npm**, that you'll use to install other tools required for blockchain development. It helps in **managing dependencies** like Truffle and Ganache, which we'll use later.

Installation and Verification of Node.Js:

Download and install the Node.js prebuilt installer from nodejs.org (suitable for both macOS & Windows) as shown in this figure, **by visiting this website** -> <https://nodejs.org/en/download/prebuilt-installer>



To verify: Run these commands “node -v” , and “npm -v” The version of node, npm installed on your system will be displayed on your terminal.

```
[192:~ apple$ node -v
v16.15.1
[192:~ apple$ npm -v
8.11.0
```

Ganache:

Ganache creates a local blockchain on your computer, where you can test your smart contracts and interactions without using real money. This will be your "fake" Ethereum network where you can deploy contracts and run transactions safely.

Installation Steps for Ganache:

Open the terminal and run “**npm install -g ganache-cli**” since this command requires edit access to globally install ganache this might throw an error because by default you are not allowed to edit it.

```
192:~ apple$ npm install -g ganache-cli
npm WARN config global '--global', '--local' are deprecated. Use '--location=global' instead.
npm ERR! code EACCES
npm ERR! syscall rename
npm ERR! path /usr/local/lib/node_modules/ganache-cli
npm ERR! dest /usr/local/lib/node_modules/.ganache-cli-NPDRaG9s
npm ERR! errno -13
npm ERR! Error: EACCES: permission denied, rename '/usr/local/lib/node_modules/ganache-cli' -> '/usr/local/lib/node_modules/.ganache-cli-NPDRaG9s'
npm ERR! [Error: EACCES: permission denied, rename '/usr/local/lib/node_modules/ganache-cli' -> '/usr/local/lib/node_modules/.ganache-cli-NPDRaG9s'] {
  errno: -13,
  code: 'EACCES',
  syscall: 'rename',
  path: '/usr/local/lib/node_modules/ganache-cli',
  dest: '/usr/local/lib/node_modules/.ganache-cli-NPDRaG9s'
}
npm ERR! The operation was rejected by your operating system.
npm ERR! It is likely you do not have the permissions to access this file as the current user
npm ERR! If you believe this might be a permissions issue, please double-check the
npm ERR! permissions of the file and its containing directories, or try running
npm ERR! the command again as root/Administrator.

npm ERR! A complete log of this run can be found in:
npm ERR! /Users/apple/.npm/_logs/2024-10-28T14_46_55_385Z-debug-0.log
```

Run this command “**sudo npm install -g ganache-cli**” in that case.

After installing Ganache, **run the command “ganache-cli**”, this will create a local blockchain, and generate a set of fake accounts and private keys.(**launching the local blockchain**)

```
Ganache CLI v6.12.2 (ganache-core: 2.13.2)

Available Accounts
=====
(0) 0x1B81c561A56ECb4339F48AC3146A9e164F3fC7D (100 ETH)
(1) 0x8067196948c5748064e8b0c5ebc1b1485f8AB39 (100 ETH)
(2) 0x5FE4E6f041cF7567bd138365AE5FAB8E203257D (100 ETH)
(3) 0xf02C2De725C8aF66A7a9099b8c47f19d17846A65 (100 ETH)
(4) 0xd0C182c34777Ac1670b0455232038da7808F21a (100 ETH)
(5) 0x66A95d8268FEBdd14610258d782A0dEb3C8D95a2 (100 ETH)
(6) 0x644358Ff9e9062916154d3f8E00F-c8Ca95b244 (100 ETH)
(7) 0x9E6A4E95bd268E21fBac93b657993655627753EE (100 ETH)
(8) 0xd3C9188384aeD091540F131D5463C4377308AB8a (100 ETH)
(9) 0xfC4f4895Fbc5132A8f44647b1627f45ee19D46fA (100 ETH)

Private Keys
=====
(0) 0x6e4d4d6d6d561e59e298ef7e4ab8f02c0fdbc52610c3a7277e5449a09ed963f9d
(1) 0xf3ec8c8e5ae09802e844c3ef27792266a232b8aca54ced31577b30486410a
(2) 0x806c8ee4c7b3e15b2793ea7d0ddfd1a7d0ef4e158092316ac1cb066557f446
(3) 0x021643bd787178c2d002d4d350ac26661cc5e20a5e26bd18a2268e81d4e9f9b
(4) 0xd439846fcc8e926fc2c2003e8896a300b0de83afe41183307459a1069e1954
(5) 0x1c8058207ad7c1c505732fc6e9fde0cbf3fe559066951d46d769e0f4e17030c
(6) 0xc94d34485014b77fdeb3bea915c25d20e99a2efe51a5c71a42789bclaf3333
(7) 0x7d6dcb004af4e0128aa3b73aabd6c220d5c7503f063f2ca7ae7375d0bd24c4
(8) 0x594e1ab537d8e82ca09957833b47e28d04705dee9927ac4017f1bcd25222dc
(9) 0xf8180bfc41d04dec235e39def2c2b42d0d3d27d4292c2b8e6e9351cecec153c

HD Wallet
=====
Mnemonic: glide current math unit walnut useless dilemma daughter language saddle nest night
Base HD Path: m/44'/60'/0'/0/{account_index}

Gas Price
=====
70000000000

Gas Limit
=====
6721975

Call Gas Limit
=====
9007199254740991

Listening on 127.0.0.1:8545
eth_blockNumber
```

TRUFFLE:

Think of Truffle as the foundation for your project, a framework that simplifies the entire development workflow for smart contracts on Ethereum. Truffle helps you manage the complexities of building a decentralized application (dApp), allowing you to focus on compiling, deploying, and testing smart contracts.

Installation Steps for Truffle:

Run the command “**npm install -g truffle**” and verify the installation by running this command in the terminal “**truffle version**”

```
C:\Users\ANKIT\Election>npm install -g truffle
```

```
C:\Users\ANKIT\Election>truffle version
Truffle v5.11.5 (core: 5.11.5)
Ganache v7.9.1
Solidity v0.5.16 (solc-js)
Node v20.16.0
Web3.js v1.10.0
```

Truffle Commands:

1. **truffle init** : This command sets up a new Truffle project structure with folders like **contracts**, **migrations**, and **test**. Useful for any new project.
2. **truffle compile**: Compiles all the Solidity contracts in the **contracts** folder and generates necessary files (ABI, bytecode) in the **build/contracts** folder.
3. **truffle migrate**: Runs all migration scripts in the **migrations** folder to deploy the contracts to a configured blockchain (local, testnet, or mainnet).
4. **truffle develop**: Starts a local blockchain (similar to Ganache) and provides a console for testing and deploying contracts without external dependencies.

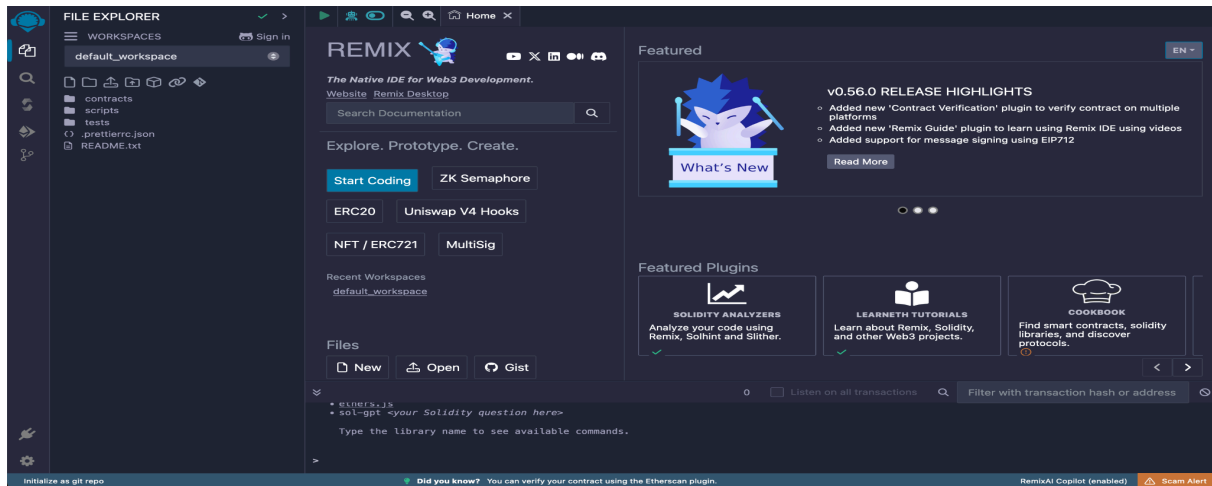
REMIX IDE:

Remix IDE is a powerful, web-based tool for writing, testing, and deploying smart contracts on the Ethereum blockchain. It provides a code editor, a Solidity compiler, and a deployment environment, making it ideal for beginners and advanced users to quickly develop and test blockchain applications. Remix also supports plugins for added functionality, such as debugging and automated testing, making it versatile for learning and prototyping.

Accessing Remix IDE:

You can access Remix IDE directly from your browser without installation.

Visit the official website here: [REMIX IDE](https://remix-ide.io/)



MetaMask:

MetaMask is a **cryptocurrency wallet** and gateway for interacting with blockchain networks, primarily Ethereum and other Ethereum-compatible chains. It allows users to manage their private keys, store and transfer cryptocurrencies, and interact with decentralized applications (dApps) directly from a web browser or mobile device.

MetaMask functions as a browser extension and a mobile app, creating a secure connection between a user's browser (or mobile app) and the Ethereum blockchain. It injects a **web3** instance (or **ethereum** provider object in newer versions) into the browser, enabling JavaScript applications to access blockchain features, including sending transactions, calling smart contracts, and checking account balances.

Installation Steps for MetaMask Extension:

1. Open your preferred browser (Chrome recommended).
2. Visit the Chrome Web Store and search type "MetaMask" and locate the official MetaMask extension.
3. Click on “**Add this extension**”
4. Once installed, pin the MetaMask icon to your taskbar by clicking the puzzle piece icon (Extensions) in the top right corner of your browser, then click the pin next to MetaMask. This always ensures easy access to MetaMask.

Create Your MetaMask Account:

1. After installing, click the MetaMask icon in your browser's taskbar to open it.
2. Follow the on-screen instructions to create a new account
3. During the setup, **MetaMask will generate a 12-word Secret Recovery Phrase (SRP)**.
 - a. **This phrase is extremely important**—it is the **only way to recover your account** if you forget your password or lose access to your device. Write down your Secret Recovery Phrase on a piece of paper and store it in a secure place. Once you have securely stored your Secret Recovery Phrase, follow the remaining prompts to complete the account creation process. After setup, MetaMask will take you to your wallet interface as shown in the figure, where you can start interacting with decentralized applications (dApps).

