1) Write about solidity Compiler.

Solidity is an object-oriented language especially developed for contract writing. It is a high-level language, which inherits traits from C++, Python and JavaScript. The solidity compiler compiles your source code into byte code that runs on Ethereum Virtual Machine (EVM):

Syntax:

```
Pragma solidity >=0.4.22 <0.6.0; // It is a directive
                                    to the compiler.
contract Ballot {           // Starts the definition of the
                                                   contract
  address chairperson;  // variables
  constructor (uint8 numProposals) public  // constructor
  {
    function giveRight to Vote (address tovoter) public //method
    {
    }
  }
}
```

The public keyword makes this method publicly invokable by any client who has access to the contract.

=> The sample contract defines three more methods called delegate, vote and winning proposal.

Ex: Contract myContract {
    unit amount
    unit Value

```
Constructor (unit initialAmount, unit initialValue) Publ
{
    amount = 0;
    Value = 1000;
}
function getBalance () public view returns (uint){

        return Value;
}
function getAmount () public view returns (uint)
{
    return amount;
}
function send (uint newDeposit) Public {

        Value = Value - newDeposit;
        amount = amount + newDeposit;

}
}.
```

Error types :-

1) JSON Error :- JSON input doesnot confirm to the required format, eg :- input is not a JSON object, the language is not supported, etc.....

2) IO error :- IO and import processing errors, such as unresolvable URL or has mismatch in supplied sources.

3) Parse error :- source code doesn't conform to the language rules.

4) Doc string Parsing error :- The net spec tags in the command block can not be parsed.

5) syntax error:

syntactical error, such as continue is used outside of a for loop.

Declaration Error, Type error, unimplemented feature error, internal compile error, Exception, compile error, fatal error, warnings.

2) write about Ganache.

→ Ganache is a personal blockchain for ethereum development you can use to deploy contracts, develop your applications and run tests.

It is available as both a desktop application as well as a command-line tool (formerly known as the test RPC).

→ Ganache is available for windows, Mac and linux.

→ It gives you the ability to perform all actions you would on the main chain without the cost.

→ It provides convenient tools such as advanced mining controls and a built-in-block explorer.

3) write about Metamask?

→ Metamask is a bridge that allows you to visit the distributed web at tomorrow in your browser today.

→ It allows you to run ethereum dApps right in your browser without running a full ethereum node.

→ Metamask includes a secure identity vault, providing

a user interface to manage your identities on diff-
-rent sites and sign block chain transactions.

→ Metamask is a cryptocurrency wallet which can be used on the chrome, firefox & Brave browsers. It's also a browser extension means it works like a bridge. b/w normal browsers and the ethereum block chain

→ It stores the ether coins and tokens based on ethereum platform. eg: EC20.

4) write about truffle?

→ Truffle is a development environment, testing frame work and asset pipeline for ethereum, aiming to make life as an ethereum developer easier, with truffle you get:

→ Built-in smart contract compilation, linking, deploym-
-ent & binary management.

→ Automated contract testing with mocha & chai.

→ Configurable build pipeline with support for custom build processes.

→ Scriptable deployment & migrations frame work.

→ Network management for deploying to many public & private networks.

→ Interactive console for direct contract communication

→ External rebuilding development

→ external script runner that executes scripts with in a truffle environment.

<u>Install</u>:

```
$npm install -g truffle
```

<u>quick usage</u>:

```
$truffle init
```

from there, you can run truffle Compiler, truffle migrate and truffle test to compile contracts, deploy those contra-cts to the network and run their associated unit tests.

→ ganache_cli :- a command_line version of truffle's blockchain server

→ ganache : A GUI for the server that displays your transaction history and chain rule.

5) write about Remix?

Remix is a suite of tools to interact with the ethereum block chain in order to debug transactions, stored in this Git.repository. A Remix transaction web debugger is available here and it source code is part of this repository.

→ The Remix IDE is an IDE for solidity App developer powered by Remix. The Remix IDE Repository is available and online.

How to use Remix :

Pre-requisites:

To use Remix tools, you will need to connect to an ethereum node.

using : geth : geth --rpc --rpcapi web3, eth, debug' --
rpcport 8545

using : eth ; eth -j --rpccorsdomain '*'.

Run - the debugger :

The debugger itself contains several controls that allow stepping over the trace and seeing the current state of a selected step.

Remix - modules :

→ Remix - analyzes
→ Remix - solidity : Provides solidity analysis and decoding functions
→ remix - lib
→ remix - debug : allow debugging transaction.
→ remix - tests : providers a uint testing for solidity
→ remix - ast walker : Provides a tool for parsing solidity AST
→ remix - url - resolver
→ remixd.