**Q: What is truffle?**

A: Truffle is a world-class development environment, testing framework and asset pipeline for blockchains using the Ethereum Virtual Machine (EVM), aiming to make life as a developer easier.

* It is considered as the most popular tool for blockchain application development with over 1.5 million lifetime downloads.
* It has a built-in smart contract compilation, linking, deployment and binary management.
* It has a built-in interactive console for direct contract communication.
* Automated contract testing for rapid development.

**Q: What are the features of truffle?**

A:

Smart Contract Lifecycle Management: - It includes support for custom deployments, library linking and complex Ethereum applications

Automated Contract Testing: - It can write automated tests for our contracts in both JavaScript and Solidity, and helps in developing contracts quickly.

Scriptable Deployment & Migrations: - Migrations are JavaScript files that helps deploying contracts to the Ethereum network. These files are responsible for staging your deployment tasks, and they're written under the assumption that the developers deployment needs will change over time.

The history of previously run migrations is recorded on-chain through a special Migrations contract which helps save the developers work.

**Q: What is blockchain?**

A: A blockchain is a permanent, unalterable, secure, and transparent ledger of transactions that is updated in real time and distributed across a peer-to-peer network of nodes. Each transaction includes a timestamp and encrypted identification information. Anything added to the platform can be made public or kept private and users can choose when to transact it on the blockchain

**Q: What is the difference between blockchain and database?**

A: Blockchain is a distributed ledger technology that is used to record transactions across many computers so that the record cannot be altered retroactively without the alteration of all subsequent blocks and requires no central authority. A database is a collection of information that is organized so that it can be easily accessed, managed, and updated.

**Q: What is NodeJS?**

A: Node.js is an open-source, cross-platform, JavaScript runtime environment that executes JavaScript code outside a web browser.

**Q: Why Do We Use NodeJS?**

Ans:

* NodeJS is built on Google Chrome’s V8 engine, and for this reason its execution time is very fast and it runs very quickly.
* There are more than 50,000 bundles available in the Node Package Manager and for that reason developers can import any of the packages any time according to their needed functionality for which a lot of time is saved.
* Node.js saves a lot of time because the files are processed and uploaded simultaneously by Node.js So as a result, the overall peed of data and video streaming is improved by NodeJS.

**Q: What is NPM?**

A: It is a command line tool that installs, updates or uninstalls Node.js packages in our application. The entire node community around the world creates useful modules and publishes them as packages in this repository.

Npm has 2 main functionalities:

1. Provides online repositories for node.js packages/modules which are searchable on the official site search.nodejs.org
2. Command line utility to install Node.js packages, do version management and dependency management of Node.js packages.

NPM comes bundled with Node.js installables after v0.6.3 version. To search for its version input *npm --version* in the command line to see the version.

**Q: What is reactjs?**

A: React is a JavaScript library for building user interfaces. It is maintained by Facebook and a community of individual developers and companies. React can be used as a base in the development of single-page or mobile applications.

**Q: Why use reactjs?**

A:

* React is used to build single page applications which makes the website looks smooth and connected.
* React allows us to create reusable UI components.
* React was first deployed on Facebook's newsfeed in 2011 and on Instagram.com in 2012.
* React automatically installs and creates all the files for you.

**Q: What are Dapps?**

A: A Dapp is also known as Decentralized Application.

It is an application built on a decentralized network that combines a smart contract and a frontend user interface

These decentralized applications are built on the open-source, peer-to-peer network of Ethereum Blockchain. Additionally, on Ethereum smart contracts are accessible and transparent just like open APIs.

A dapp can have frontend code and user interfaces written in any language. Its nature of isolation (dapps are executed in a virtual environment known as Ethereum Virtual Machine so that if the smart contract has a bug, it won’t hamper the normal functioning of the blockchain network), deterministic (dapps perform the same function irrespective of the environment in which they get executed) and decentralized comes quite handy during development.

**Q: Benefits of dapp development**

A:

**Zero downtime –** Once the smart contract is deployed on the blockchain, the network as a whole will always be able to serve clients looking to interact with the contract. Malicious actors, therefore, cannot launch denial-of-service attacks targeted towards individual dapps.

**Privacy –** There's no need to provide real-world identity for deploying or interacting with a dapp.

**Resistance to censorship –** No single entity on the network can block users from submitting transactions, deploying dapps, or reading data from the blockchain.

**Complete data integrity –** Data stored on the blockchain is immutable and indisputable. Malicious actors cannot forge transactions or other data that has already been made public.

**Trustless computation/verifiable behaviour –** Smart contracts can be analysed and are guaranteed to execute in predictable ways, without the need to trust a central authority. This is not true in traditional models; for example, when we use online banking systems, we must trust that financial institutions will not misuse our financial data, tamper with records, or get hacked.

**Q: Drawbacks on dapp development**

A:

**Maintenance –** Dapps can be harder to maintain because the code and data published to the blockchain are harder to modify. It’s hard for developers to make updates to their dapps once they are deployed, even if bugs or security risks are identified in an old version.

**Performance overhead –** There is a huge performance overhead, and scaling is really hard. To achieve the level of security, integrity, transparency, and reliability that Ethereum aspires to, every node runs and stores every transaction. On top of this, proof-of-stake consensus takes time as well.

**Network congestion –** When one dapp uses too many computational resources, the entire network gets backed up. Currently, the network can only process about 10-15 transactions per second; if transactions are being sent in faster than this, the pool of unconfirmed transactions can quickly balloon.

**User experience –** It may be harder to engineer user-friendly experiences because the average end-user might find it too difficult to set up a tool stack necessary to interact with the blockchain in a truly secure fashion.

**Q: What is Solidity?**

A: Solidity is a contract-oriented, high-level language for implementing smart contracts. It is designed to target the Ethereum Virtual Machine (EVM). Solidity is statically typed, supports inheritance, libraries and complex user-defined types among other features.

**Q: Why solidity?**

A: Solidity is an object-oriented programming language created specifically by the Ethereum Network team for constructing and designing smart contracts on Blockchain platforms.

It's used to create smart contracts that implement business logic and generate a chain of transaction records in the blockchain system.

It acts as a tool for creating machine-level code and compiling it on the Ethereum Virtual Machine (EVM).

It has a lot of similarities with C and C++ and is very simple to learn and understand. For example, a “main” function in C is equivalent to a “contract” in Solidity.

**Q: What is an Ethereum Virtual Machine?**

A: A EVM provides a runtime environment for Ethereum smart contracts.

It is primarily concerned with ensuring the security and execution of untrusted programs through the use of an international network of public nodes.

EVM is specialized in preventing Denial-of-Service attacks and certifies that the programs do not have access to each other's state, as well as establishing communication, with no possible interference.