**Indian Web Browser Development Challenge #Day1 25/8/23**

**Basic Browser related information and trust store**

**Q1. List all browsers which are open source and have minimum one unique feature.**

1. **Firefox** – very strong privacy features
2. **Water fox** – Default Private Search, Container Tabs, Private Tabs
3. **Vivaldi** – highly customizable, notetaking system (built in VPN and ad blocker)
4. **Brave** – very fast (built in ad blocker)
5. **Chromium** – very huge collections of extensions and compatibility (very fast as not many features)
6. **TOR** 
   1. Privacy – hides IP and encrypt traffic
   2. Security – Uses onion routing
   3. Anonymity

**Q2. What are the programming languages used by each open-source browser and why they use that specific language?**

**C++:** Core of the browser (rendering pages and handling network requests)

**JavaScript**: user interface and for web development tools

**Rust:** sandboxing and memory safety

**Python:** scripting and automation tasks

**Web Assembly:** rub high-performance code in the browser

**Qt:** Used for Graphical user interface

* Firefox: C++, JavaScript, Rust
* Water fox: Same as Firefox.
* Brave: Same as Firefox.
* Vivaldi: C++, JavaScript, Python
* Chromium: C++, JavaScript, Web Assembly
* Tor Browser: C++, JavaScript, Qt

**Q2. How we make our own trust store what are the requirements?**

1. List of Trusted CA
2. Checking the intermediate CA certificate which is used to sign the website certificate
3. Checking if intermediate CA is signed by TRUSTED CA

**Q4. What is CCA India root certificate?**

1. CCA India Root Certificate is a certificate issued by the Controller of Certifying Authorities (CCA) in India.
2. It is used to sign certificates that are issued by licensed Certifying Authorities (CAs) in India.
3. It helps to ensure that users can trust the certificates that are issued by licensed CAs in India.

**Web3.0 and decentralised browsers**

**Q5. What is web 3.0?**

* Web3 is the third generation of the internet that is built on top of blockchain technology.
* **Decentralization**: Web3 is not controlled by any single entity,
* **Security**: Web3 uses cryptography to secure data and transactions, making it more secure than traditional web applications.
* **Openness**: Web3 is an open-source platform, which means that anyone can contribute to its development.
* **Interoperability**: Web3 applications can communicate with each other.

**Q6. What are the advantages of web 3.0?**

* **More secure and transparent data ownership:** Web3 uses blockchain technology to store data. This is because blockchain is a decentralized ledger that is maintained by a network of computers, making it very difficult to hack or tamper with data.
* **Increased user privacy:** Web3 uses cryptography to protect user data, making it more difficult for third parties to track and collect user data.
* **More democratic and equitable internet:**  it is not controlled by any single entity
* **New and innovative applications:** interoperable, decentralized application

**Q7. What are the things required to support web3.0?**

* **Blockchain technology**: A decentralized ledger that is maintained by a network of computers.
* **Cryptography**: The use of codes to secure user data and transactions.
* **Decentralized applications (dApps**): Applications that are not controlled by any single entity.
* **Smart contracts:** Self-executing contracts that are stored on the blockchain.
* **Interoperability:** The ability of applications to communicate with each other.

**Q8. What is the relation between web3 and blockchain?**

Web 3 works on top of blockchain technology i.e., all of its transactions (or any work done) is performed or stored on a blockchain.

**Q9. List all opensource decentralized browsers which support web3?**

* Brave
* Tor
* Subgraph Studio
* Cosmos Browser
* Swarm Browser

**Q10.What language is used to build these decentralized browsers?**

|  |  |  |
| --- | --- | --- |
| **Browser** | **Framework** | **Languages** |
| Brave | Chromium | JavaScript, C/C++ |
| Tor | Firefox | JavaScript, Python |
| Subgraph Studio | Electron | JavaScript, TypeScript, Go |
| Cosmos Browser | Substrate | Rust, JavaScript |
| Swarm Browser | Swarm | Go, JavaScript |

* **JavaScript:**  user interface and for running scripts on web pages.
* **C/C++:** core of the browser, such as the rendering engine and the networking stack.
* Python: Python is used for the core of the Tor network, such as the directory servers and the onion routing protocol.
* **Go:**
* core of Subgraph Studio, such as the GraphQL server and the data processing pipeline.
* core of Swarm Browser, such as the networking stack and the data storage layer.
* **Rust:** core of the Cosmos Browser, such as the networking stack and the consensus algorithm.

**Browser related more information: -**

**Q11. Why Chrome is ram hungry?**

* **Multi-process architecture:** each tab and extension run in its own process. This allows Chrome to be more responsive and efficient, but it also means that it uses more RAM.
* **Preload:**Chrome preloads pages in the background, even if you're not actively viewing them.
* **Heavy use of JavaScript**

**Q12. What are the things which make a browser secure?**

* **Security updates:**It should automatically update itself with the latest security patches.
* **Sandboxing:**Websites would be isolated from each other to prevent them from accessing each other's data or resources.
* **HTTPS:**It should always use HTTPS, a secure protocol that encrypts data sent between the browser and the website.
* **Cookies:**Users would be able to choose which cookies they want to allow or disable cookies altogether.
* **Ad blockers:**It should include an ad blocker to protect users from malicious ads.
* **Privacy settings:**Users would be able to adjust their privacy settings to protect their privacy.
* **Hardware-backed security:**It should use hardware, such as a TPM, to store encryption keys and other sensitive data.
* **Threat intelligence:**It should use threat intelligence to identify and block malicious websites and ads.
* **User education:**It should include features to educate users about security risks and how to protect themselves.

**Q13. Why Brave Browser is secure?**

* **Built-in ad blocker:**Blocks ads by default, which helps to protect your privacy and security.
* **Fingerprinting protection:**Makes it more difficult for websites to track you by preventing them from building a profile of your browsing habits.
* **HTTPS Everywhere:**Upgrades all possible connections to HTTPS, which is a more secure protocol than HTTP.
* **Security updates:**Regularly updated with security patches to protect against the latest threats.
* **Sandboxing:**Isolates each tab and extension from each other, helping to prevent malware from spreading.
* **Intelligent tracking prevention:**Blocks third-party cookies, trackers, and fingerprinting attempts.
* **Certificate pinning:**Verifies the identity of websites you visit, helping to protect you from man-in-the-middle attacks.
* **Security advisor:**Provides warnings about potentially dangerous websites and downloads.
* **Brave Firewall:**Blocks malicious traffic from reaching your computer.
* **Distributed ledger**: Brave uses a distributed ledger to track user data. (browsing history)

**Q14. Brave, Chrome, Edge uses chromium as their web engine then why Brave Browser is fast?**

* Brave block ads and trackers natively (no extension required) resulting in load speed (upto3x of pages)
* It doesn’t track additional user data in background which also improves speed

**Q15. What are the things required to build a minimal browser with only search functionality?**

* **Web engine:**  for rendering pages (gecko or blink
* **User interface**: to search and show rendered pages
* **Search engine**
* **Networking stack:** way to establish TCP connections, way to resolve hostnames to ip address.

**Q16. How lite browsers available on market/play store works? How they are small in size and fast also?**

* Removes non-essential features (extensions, plugins, autofill, themes, customization)
* Optimized to use less memory: releasing unused memory asap
* Uses a minimalistic user interface
* Light weight web engine (gecko view)
* Using compression (for data transferred between server and browser)
* SSR and Caching
* Reduced startup time (loading only essential things)

**By :- Devang Shaurya Pratap Singh**