

 **Connect the Dots – Ethers.js and MetaMask UI**  
  
**Objective/Aim:**  
To demonstrate how to connect a web application’s frontend to the Ethereum blockchain by integrating Ethers.js with MetaMask, enabling wallet connection, transaction signing, and blockchain interaction through a user-friendly interface.

**Apparatus/Software Used:**

* Ethers.js JavaScript library for blockchain communication
* MetaMask
* Web3-enabled browser(brave)

**Theory/Concept:**

Ethers.js is a lightweight library that facilitates Ethereum blockchain interaction by providing abstractions for providers (to read the blockchain) and signers (to sign transactions). MetaMask acts as the user’s wallet and Ethereum provider. When integrated with a dApp UI, Ethers.js helps connect the dApp to MetaMask, allowing users to authenticate, sign transactions, and query blockchain data seamlessly via the browser



**Procedure:**

* Check MetaMask Presence: Detect window.ethereum to verify that MetaMask is installed.
* Initialize Provider: Create a new ethers.providers.Web3Provider(window.ethereum) instance.
* Request Account Access: On user action (e.g., "Connect Wallet" button), invoke provider.send("eth\_requestAccounts", []) to prompt MetaMask to connect.
* Obtain Signer: Use provider.getSigner() to get the signer for sending transactions.
* Update UI: Show the connected wallet address and optionally the network and balance information.
* Send Transactions: Use signer methods to create and send transactions, with MetaMask handling user confirmation.
* Handle Events: Listen for account or network changes and update the UI accordingly.Implement peer discovery protocols where nodes identify and connect to other nodes

**Observation:**

* Users can connect their MetaMask wallets effortlessly through the web UI.
* The app reflects wallet status and address in real time.
* Transactions require user approval through MetaMask, ensuring security.
* Interaction with smart contracts or blockchain data is simplified and intuitive using Ethers.js along with the wallet UI.