

WALLET App

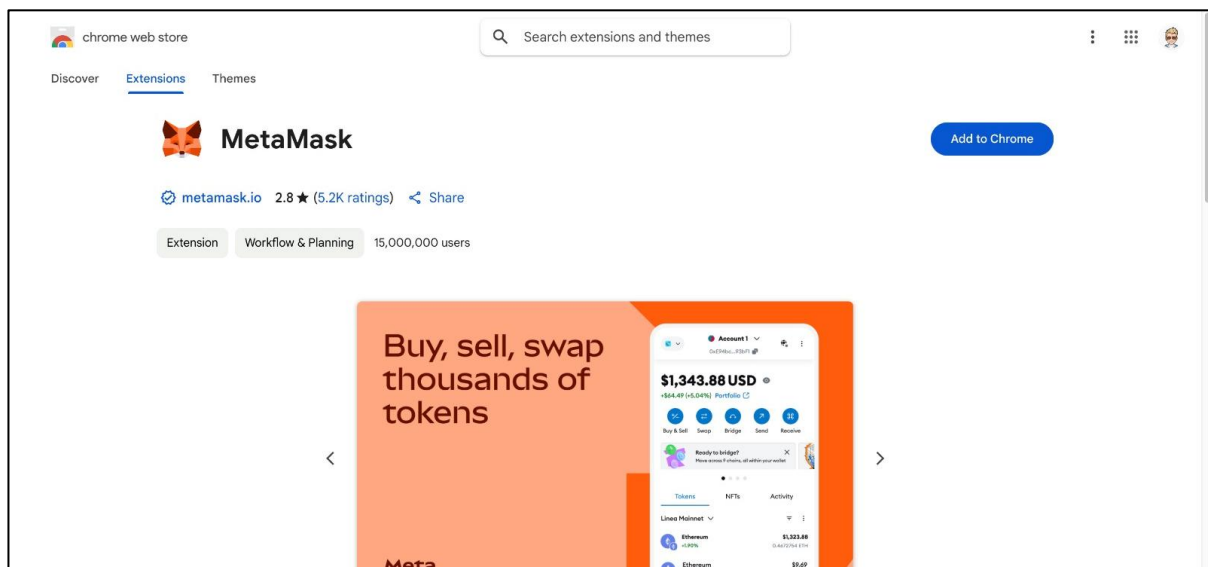
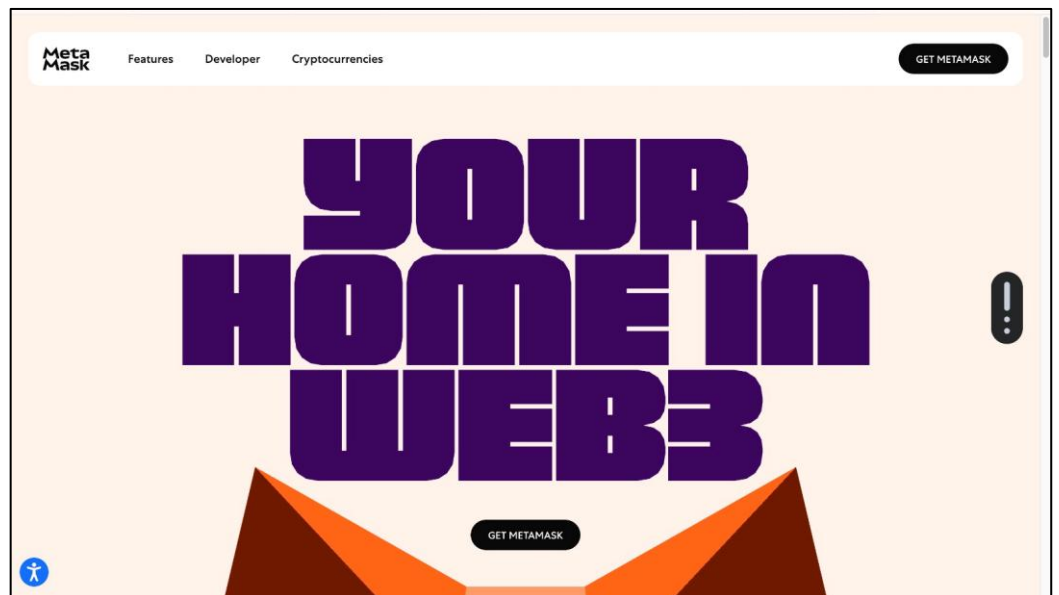
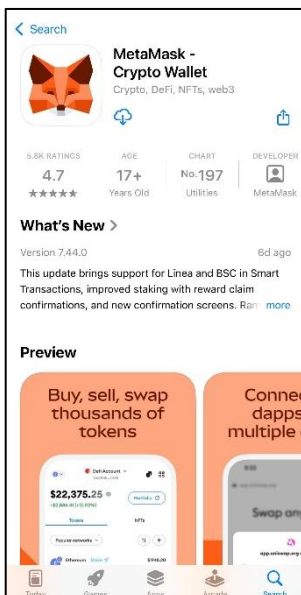
Part 1: Install MetaMask & Set Up Wallet

Step 1: Install MetaMask

1. Go to the official site: <https://metamask.io/>
2. Click **Download** and choose your browser (Chrome, Firefox or Edge).
3. Add the extension to your browser.

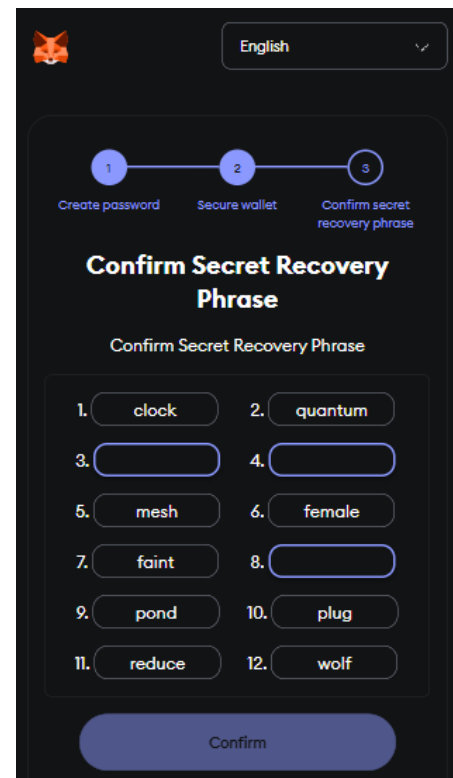
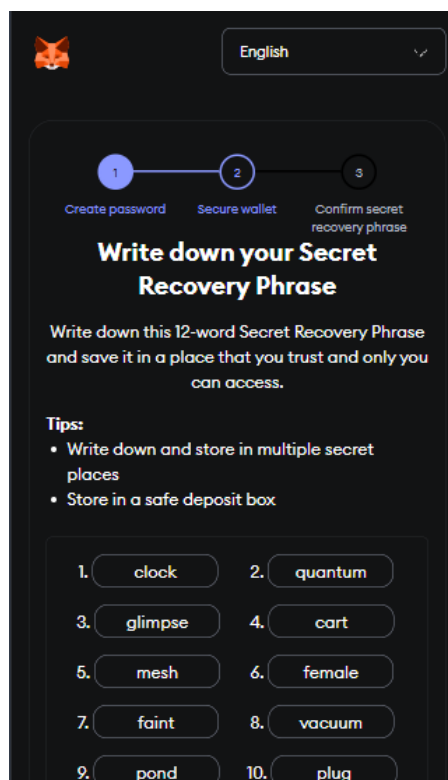
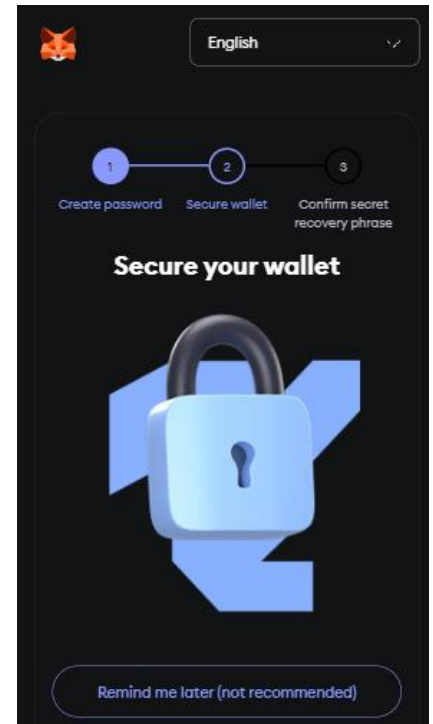
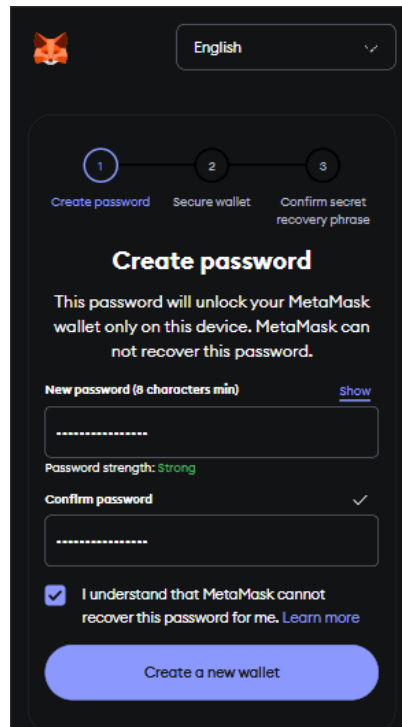
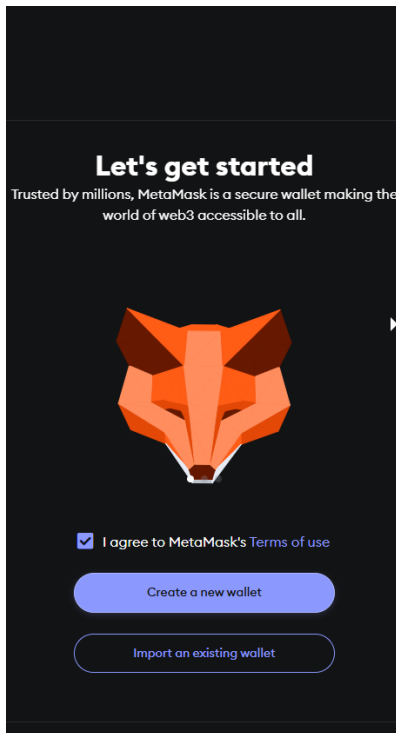
OR

1. Open the **App Store/Play Store**
2. Search for “MetaMask”
3. Tap “Install” or “Get”.



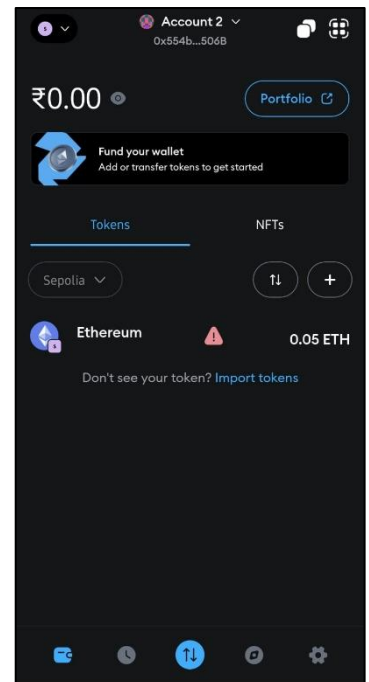
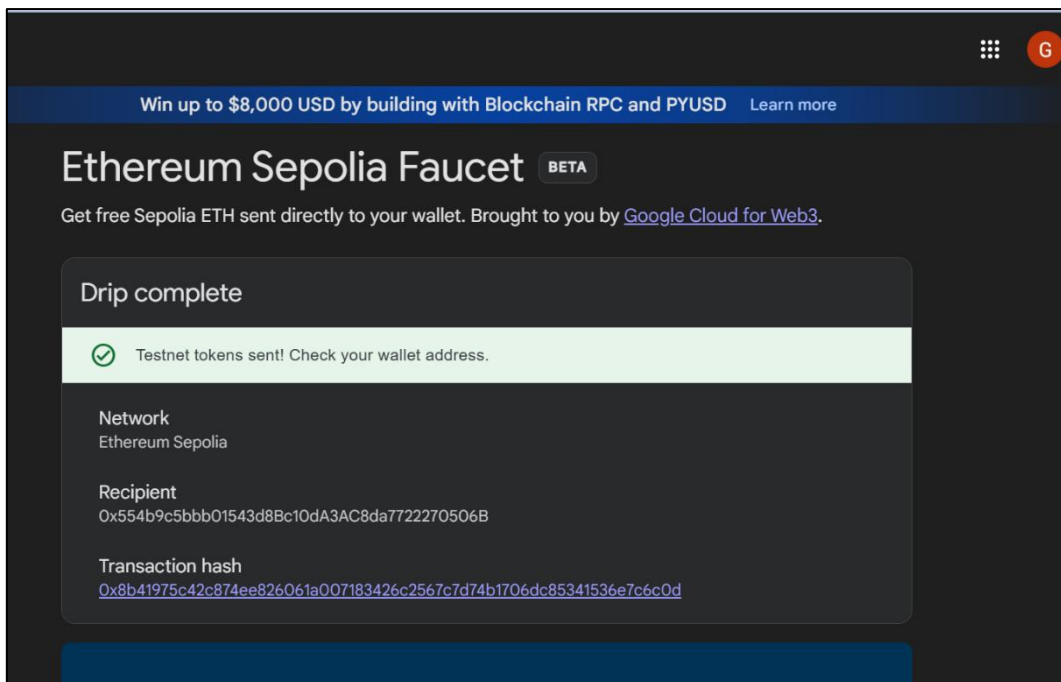
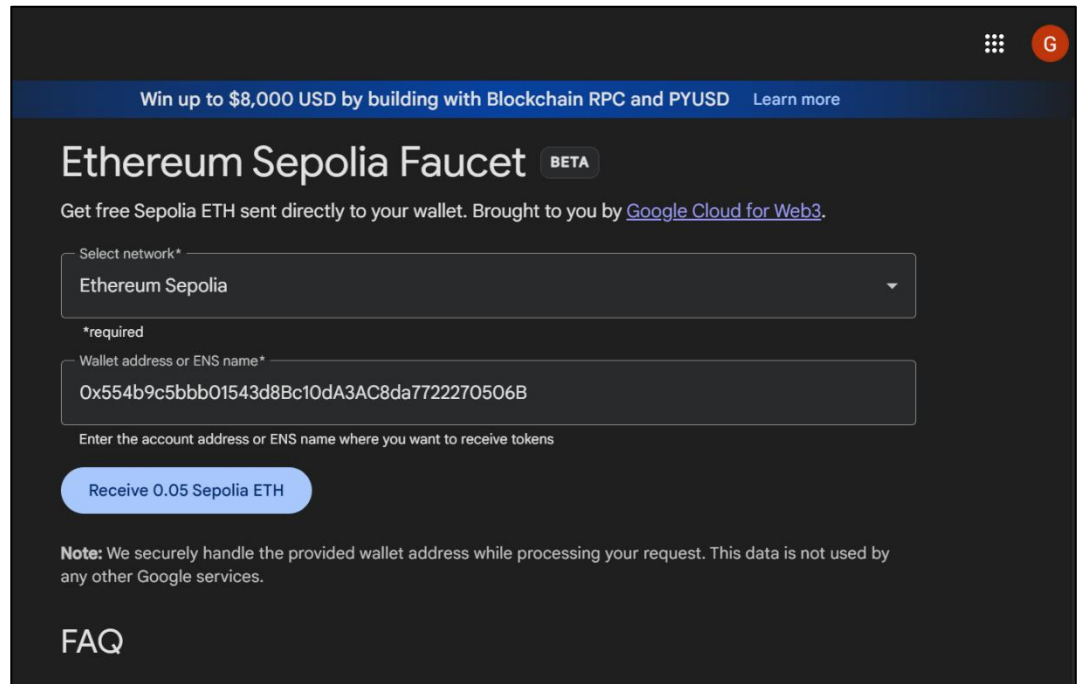
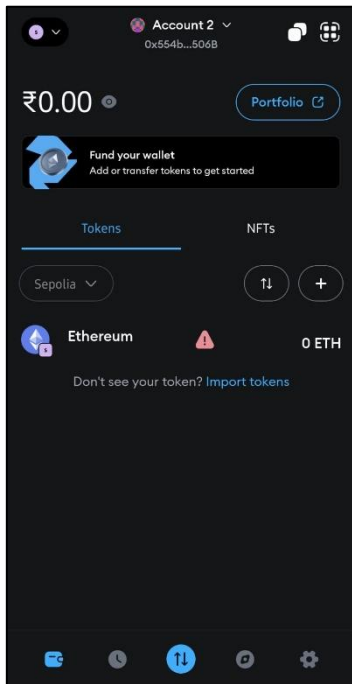
Step 2: Create or Import a Wallet

1. Click the MetaMask extension icon.
2. Choose **Create a Wallet** or **Import Wallet** (if you already have one).
3. Follow the setup steps:
 - Set a password.
 - Back up your **Secret Recovery Phrase** securely.



Step 3: Add a Test Network (Sepolia)

1. Open MetaMask.
2. Click your account icon > **Settings** > **Networks**.
3. Add a network like **Sepolia**:
4. Get free test ETH from a **faucet**:
 - Sepolia Faucet: <https://cloud.google.com/application/web3/faucet>



Part 2: Smart Contract - Ether manager App

Step 4: Code for Smart Contract

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.4;

contract EthWalletManager {
    // Event emitted when a transfer occurs
    event Transfer(
        address indexed sender,
        address indexed recipient,
        uint256 amount,
        uint256 timestamp
    );

    // Error messages
    error InsufficientBalance(uint256 available, uint256
required);
    error TransferFailed();
    error ZeroAmount();
    error ZeroAddressNotAllowed();

    function getBalance() public view returns (uint256) {
        return address(msg.sender).balance;
    }

    /**
     * @dev Transfers ETH to a specified address
     * @param recipient The address to send ETH to
     * @param amount The amount of ETH to send in wei
     */
    function transferEth(address payable recipient, uint256
amount) public payable {
        // Input validation
        if (recipient == address(0)) revert
ZeroAddressNotAllowed();
        if (amount == 0) revert ZeroAmount();

        // Check sender's balance
        if (address(msg.sender).balance < amount) {
            revert InsufficientBalance({
                available: address(msg.sender).balance,
                required: amount
            });
        }

        // Perform transfer
        (bool success, ) = recipient.call{value: amount}("");
        if (!success) revert TransferFailed();
    }
}
```

```

        // Emit event
        emit Transfer(
            msg.sender,
            recipient,
            amount,
            block.timestamp
        );
    }

    // Allow the contract to receive ETH
    receive() external payable {}
}

```

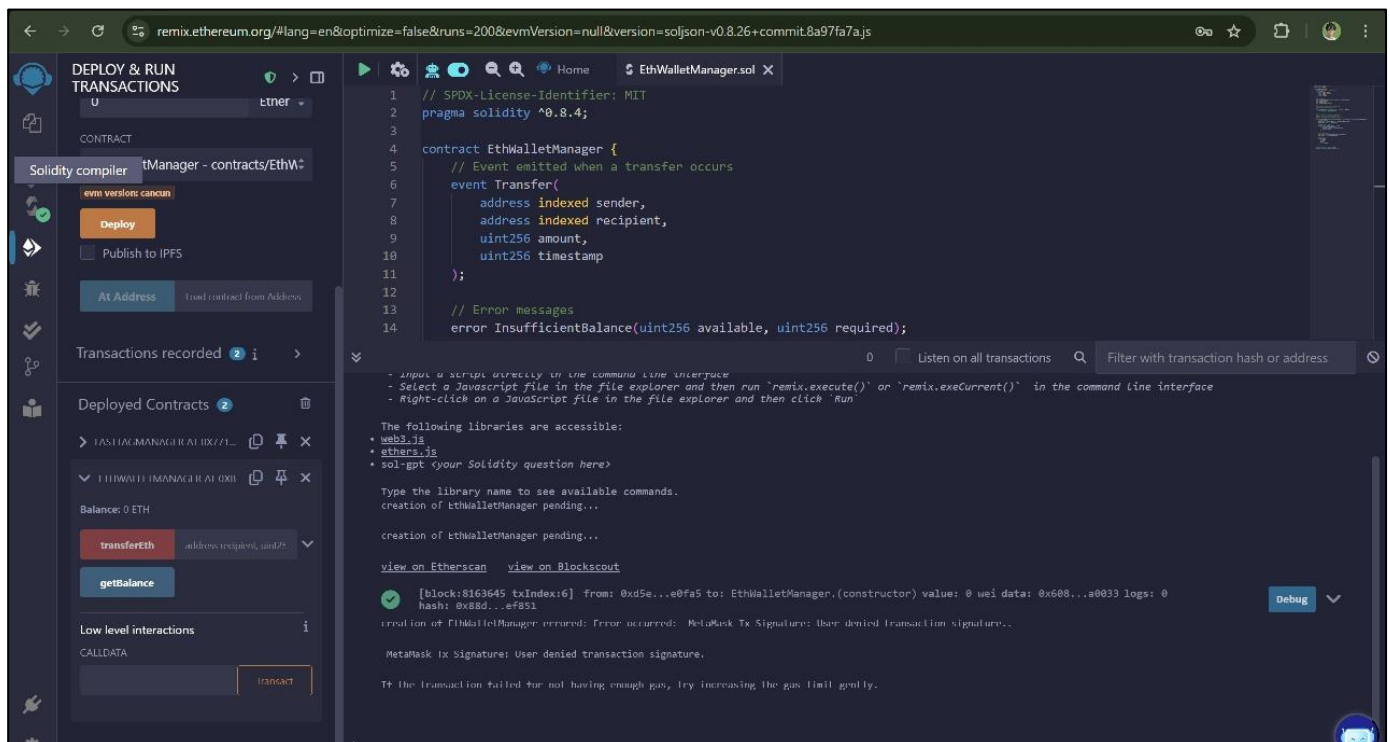
Part 3: Deploy the Contract Using Remix

Step 5: Open Remix IDE

1. Visit <https://remix.ethereum.org>
2. Create a new file and paste in your EthWalletManager contract code.

Step 6: Compile the Contract

1. Go to the **Solidity Compiler** tab.
2. Make sure the version is set to `^0.8.4` or higher.
3. Click **Compile EthWalletManager.sol**.

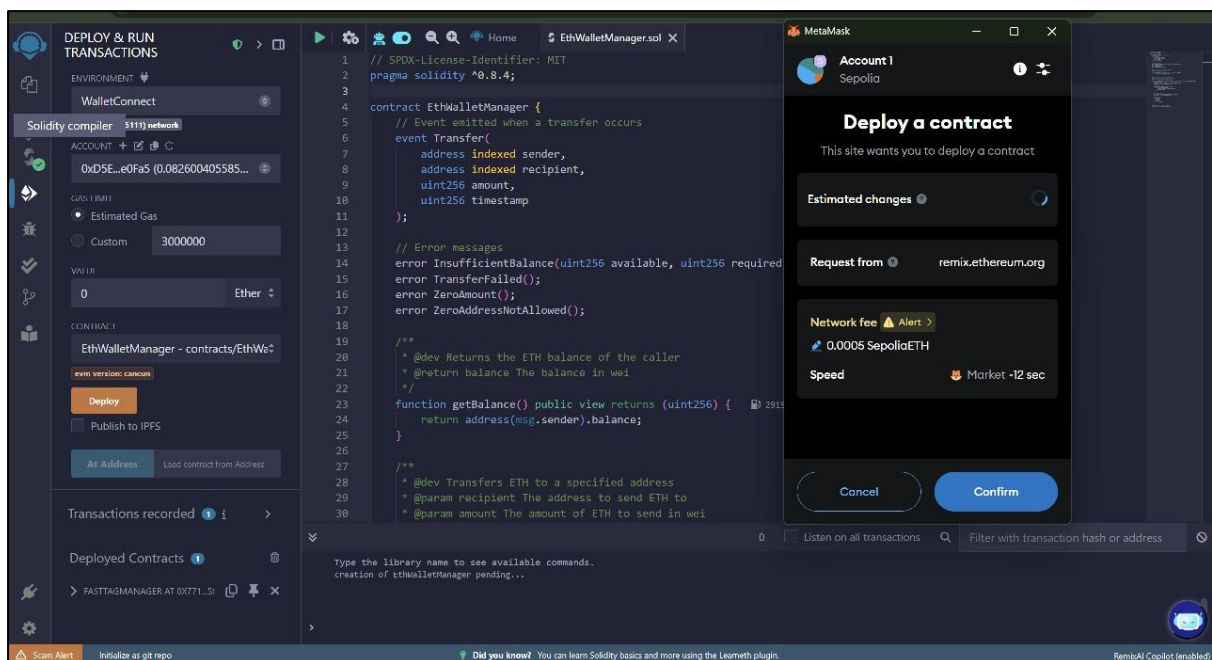


Step 7: Connect MetaMask to Remix

1. Go to the **Deploy & Run Transactions** tab.
2. In the **Environment** dropdown, choose **Injected Provider - MetaMask**.
3. MetaMask will pop up — approve the connection to Remix.

Step 8: Deploy the Contract

1. Make sure your account has some ETH (from faucet).
2. Click **Deploy**.
3. MetaMask will prompt you to confirm the transaction.
4. Once deployed, your contract address will appear in Remix.



Part 4: Interact With the Smart Contract

Step 9: Use Remix Interface

1. `getBalance()`: Calls your wallet's ETH balance.
2. Enter a **recipient address**.
3. Set an amount in **wei** (1 ETH = $1e18$ wei).
4. Make sure your wallet has enough ETH to cover the transfer.

Step 10: Interact with the Contract from a Frontend (Flutter)

1. Connect Your Wallet by Choosing **MetaMask**
2. View Your **Balance** and **Send** or **Receive** ETH
3. Optionally, tap “**View Transactions**” to see recent activity

