**CrowdFunding DApp - Project Report**

**Team Members**

| **Name** | **Registration no** |
| --- | --- |
| Mahnoor Fatima | cs211188 |
| Anum Tariq | cs211163 |
| Pirthvi Raj Khatri | cs211078 |
| Anish Acharia | cs211171 |

**1. Project Overview**

**Project Name:** CrowdFund DApp  
**Type:** Decentralized Application

**Purpose**

CrowdFund DApp is a decentralized crowdfunding platform built on the Ethereum blockchain. It enables users to contribute ETH to category-based campaigns. Funds can only be withdrawn by the campaign owner if the funding goal is met. Otherwise, contributors are eligible for refunds after the campaign deadline. The platform promotes transparency and eliminates third-party interference by enforcing smart contract logic.

**2. Objectives**

* Build a trustless and transparent crowdfunding platform.
* Accept real-time ETH contributions.
* Allow campaign creators to withdraw funds only upon achieving the funding goal.
* Provide contributors the ability to request refunds after deadlines if goals are not achieved.
* Demonstrate blockchain utility using Ethereum and Solidity.

**3. Features**

| **Feature** | **Description** |
| --- | --- |
| Contribution | Any Ethereum wallet holder can contribute ETH to a selected campaign category. |
| Withdrawal | Campaign owner can withdraw funds **only** if the campaign goal is achieved. |
| Refund | Contributors can claim refunds if the campaign goal is not met after the deadline. |
| Real-time Updates | Campaign statistics update dynamically based on smart contract state. |

**4. Tools & Technologies Used**

| **Layer** | **Technology** |
| --- | --- |
| Frontend | React.js, HTML, CSS |
| Wallet Connect | MetaMask |
| Smart Contract | Solidity (v0.8.x) |
| Deployment | Hardhat |
| Blockchain | Ethereum Test Network |
| Library | Ethers.js |

**5. Smart Contract Code (Solidity)**

The smart contract was implemented in Solidity and contains logic for contributions, withdrawals, and refunds.

**Key Functions:**

* contribute(string memory category) — Accepts ETH based on category input.
* withdraw() — Allows campaign owner to withdraw funds upon reaching the goal.
* refund() — Allows contributors to request refund after deadline if the goal is unmet.

Contract ensures that conditions are enforced through modifiers and timestamp checks.

**6. Frontend Integration**

Frontend handles wallet connection, form input, and UI rendering.

**Key Functionalities:**

* **Connect Wallet** — Uses ethers.BrowserProvider and MetaMask.
* **Contribute ETH** — Captures amount and category and sends transaction to smart contract.
* **Withdraw Funds** — Available only to campaign owner after goal.
* **Claim Refund** — Displayed only if deadline has passed and goal is not met.

**UI Highlights:**

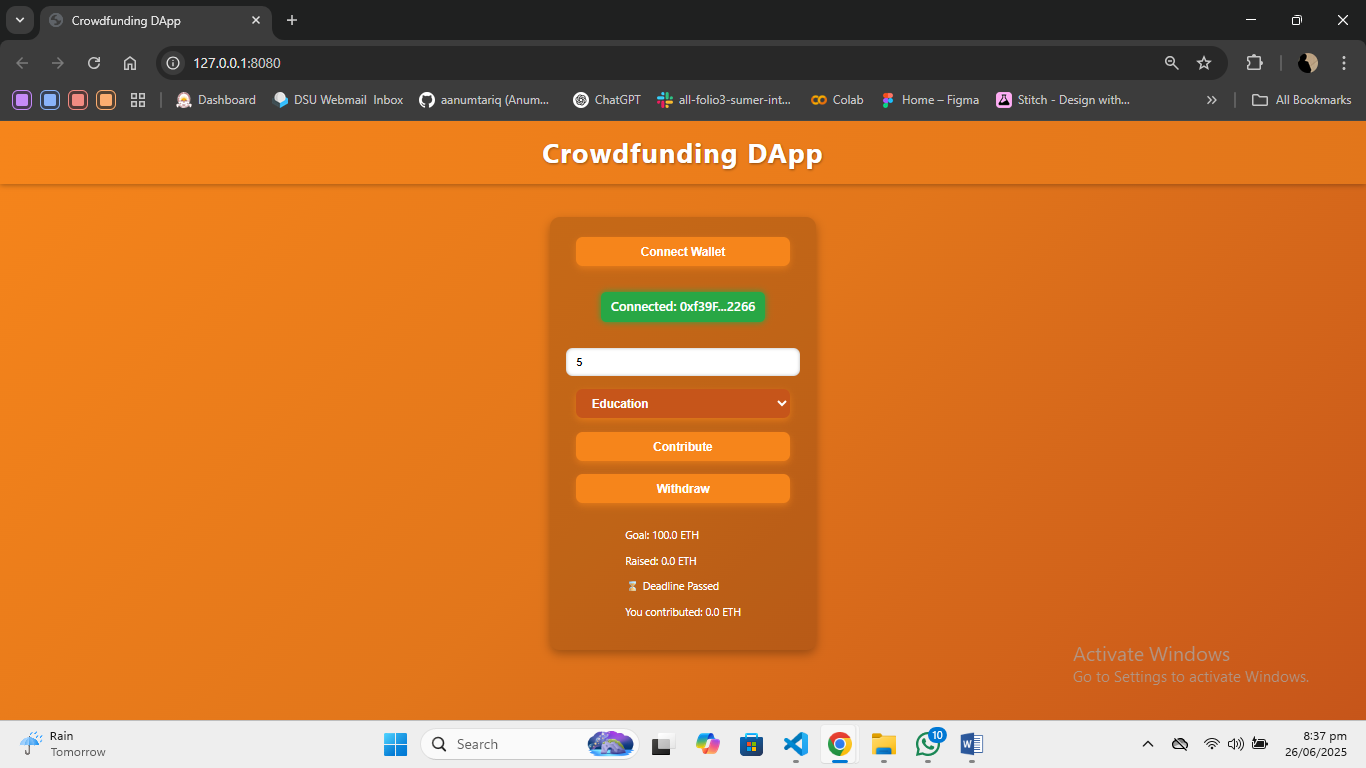
* Dynamic update of goal, raised ETH, and user's contribution.
* Time left or deadline passed message.
* Condition-based rendering of refund and withdraw buttons.

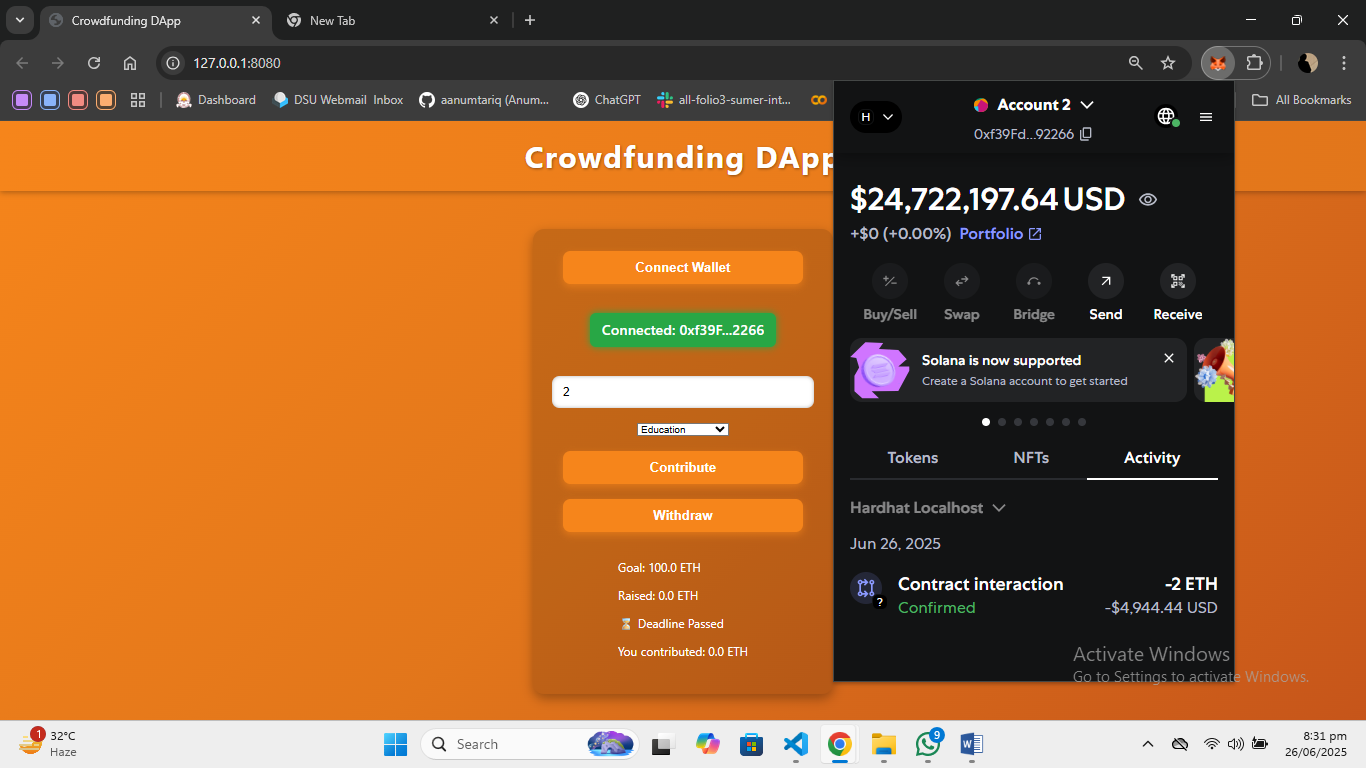
**7. Deployment Steps**

1. Developed and tested smart contract locally.
2. Used Hardhat for local blockchain simulation.
3. Deployed contract using deploy.js script.
4. Deployed to Ethereum testnet
5. Retrieved deployed contract address.
6. Integrated address into frontend React app.

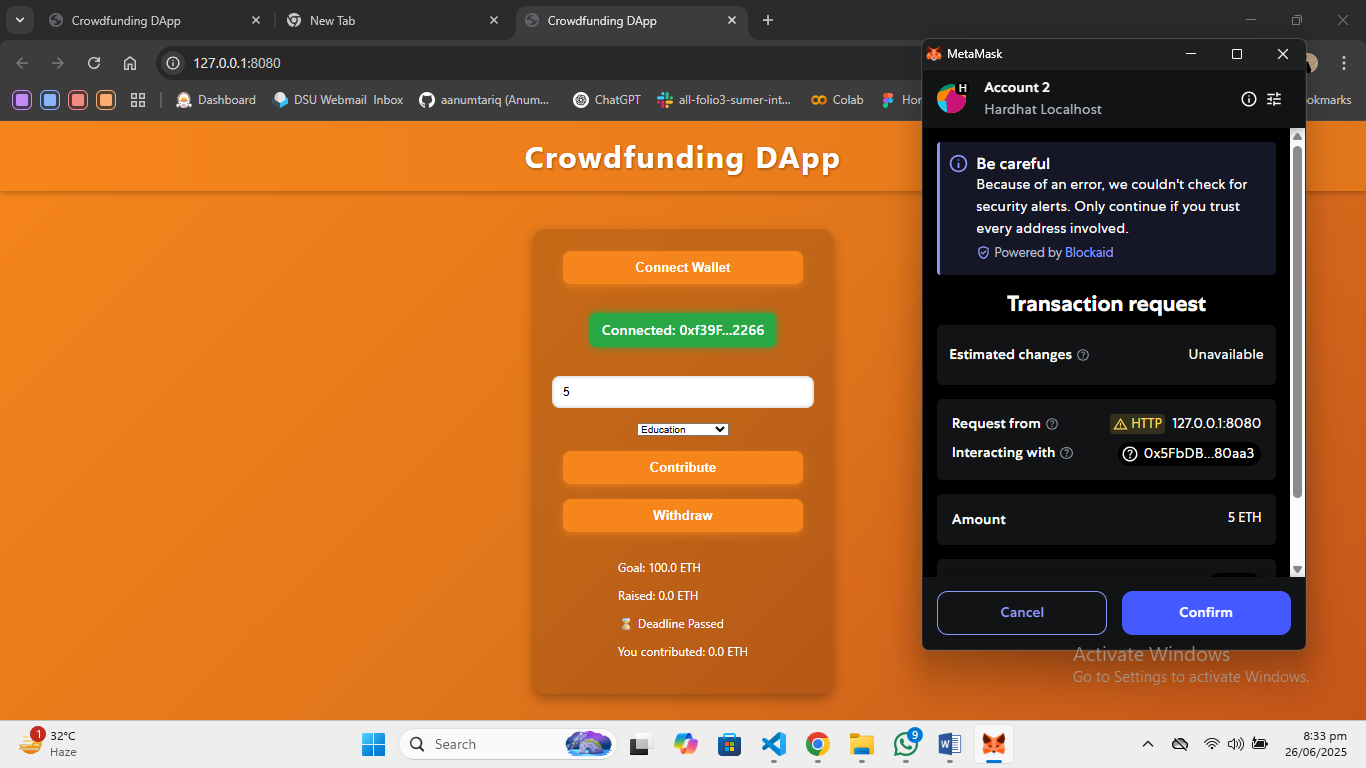
**8. Screenshots**

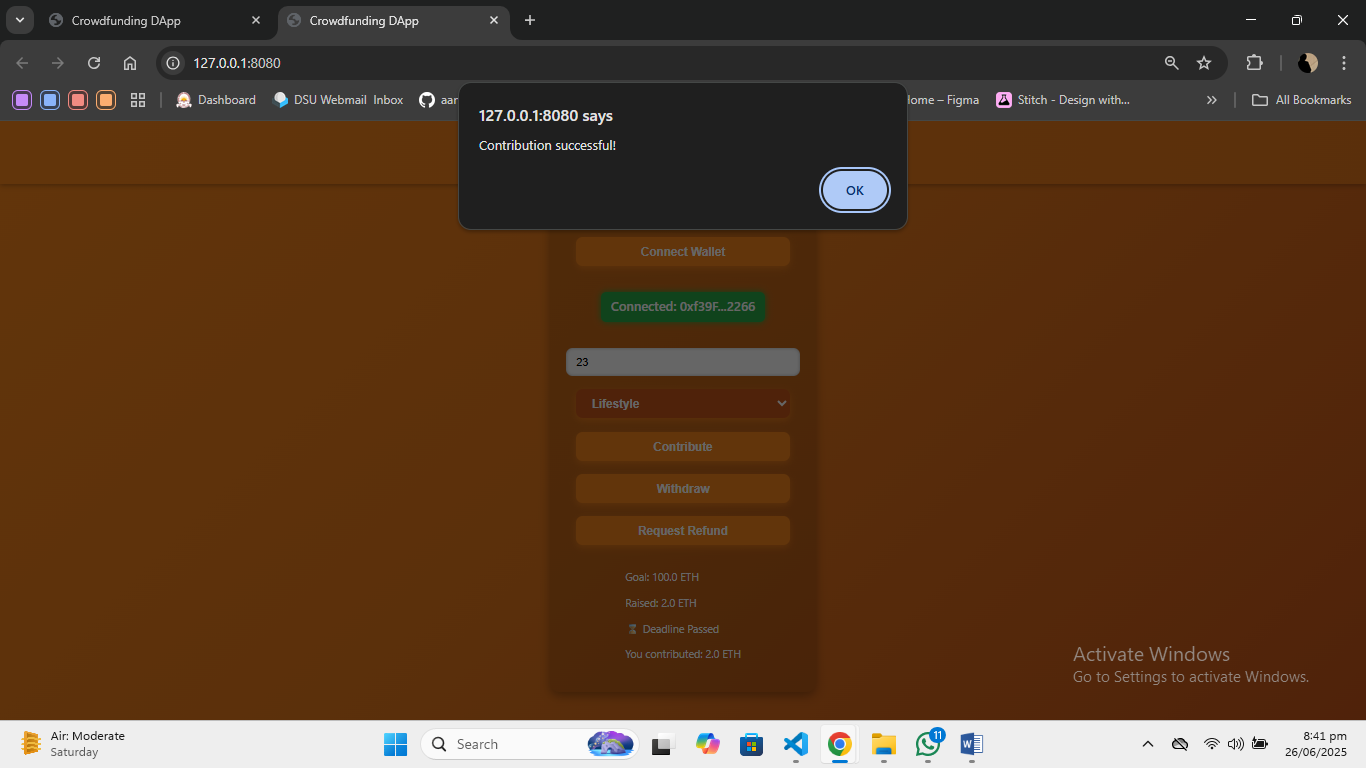
**Wallet Connection**





**Contribution Form**





**9. Conclusion**

The CrowdFund DApp project effectively demonstrates how blockchain technology can be leveraged to build a secure, decentralized crowdfunding platform. It ensures transparent fund handling, conditional logic enforcement through smart contracts, and user trust without third-party intermediaries. The project serves as a foundational step toward real-world decentralized finance (DeFi) applications.