Blockchain Development Tasks & Instructions — CodeAlpha

🛠️ Blockchain Development — Internship Overview
This internship program offers practical experience in blockchain technology and decentralized application development. CodeAlpha is a leading software development company focused on innovation, scalability, and secure architecture. The internship empowers students to build smart contracts, develop DApps and work with blockchain platforms like Ethereum, Solidity and Hyperledger. Interns will receive hands-on training, expert mentorship and opportunities to contribute to real-world blockchain projects in a collaborative environment.

🎁 Internship Perks

- Internship Offer Letter
- Completion Certificate (QR Verified)
- Unique ID Certificate
- Letter of Recommendation (based on performance)
- Job Opportunities / Placement Support
- Resume Building Support

📌 Instructions for Interns

1. Share your internship status on **LinkedIn**, tagging **@CodeAlpha**.
2. Complete the **assigned projects** within the mentioned time frame.
3. Upload your complete source code to **GitHub** in a repository named:
   `CodeAlpha_ProjectName`
4. Post a **video explanation** of your project on LinkedIn with GitHub repo link.
5. Submit your completed task using the **Submission Form**.
6. **Complete any 3 or 2 out of the 4 tasks** listed below (from your domain).

📱 Blockchain Development Task List

(Complete any 2 or 3 of the following tasks)

✅ TASK 1: Simple Storage Smart Contract

- Declare an integer variable inside the Solidity contract to store a value.
- Write an `increment` function that increases this value by 1.
- Write a `decrement` function that decreases this value by 1.
- Make sure the value can be read from outside the contract (either by making the variable public or by creating a read function).
- Compile, deploy, and test the contract to confirm both increment and decrement work correctly

---

✅ TASK 2: Multi-Send Smart Contract

- Create a Solidity contract that accepts an array of Ethereum addresses.
- Write a payable function that receives Ether with the transaction.
- Inside the function, use a loop to send an equal amount of Ether to each address in the array.
- Make sure to handle the Ether distribution correctly and check for successful transfers.
- Deploy and test the contract on Remix IDE to ensure it works as expect

---

✅ TASK 3: Polling System Smart Contract

- Create a structure for a poll with a title, options, end time, and vote count (using mappings).
- Allow users to **create polls** by providing options and a voting deadline.
- Allow each address to **vote only once** before the deadline using time-based restrictions.
- Use mappings to store votes securely and prevent double voting.
- Add a function to **determine and return the winning option** after the poll ends.

✅ TASK 4: Personal Portfolio (Crypto Locking) Smart Contract

1. Create a smart contract where users can deposit Ether or tokens along with a lock-in time.
2. Store each user's deposit amount and unlock time using mappings.
3. Use Solidity's `block.timestamp` to enforce the time-lock.
4. Create a `withdraw` function that only allows users to withdraw after the lock time has passed.
5. Deploy and test the contract on Remix IDE to verify that early withdrawals are blocked.

🚨 Important Note

Internship Completion Criteria:
To be eligible for the internship certificate, participants must complete a minimum of **two or three tasks**. Submitting only one task will be considered **incomplete** and certificates will **not** be issued in such cases.

📤 Submission Details

A submission form will be shared in your respective **WhatsApp group**. You are required to submit your completed task only through that form. Please follow the instructions mentioned in the form carefully to ensure your submission is accepted.

📞 Contact Information

- Website: www.codealpha.tech
- WhatsApp: +91 8052293611
- Email: services@codealpha.tech