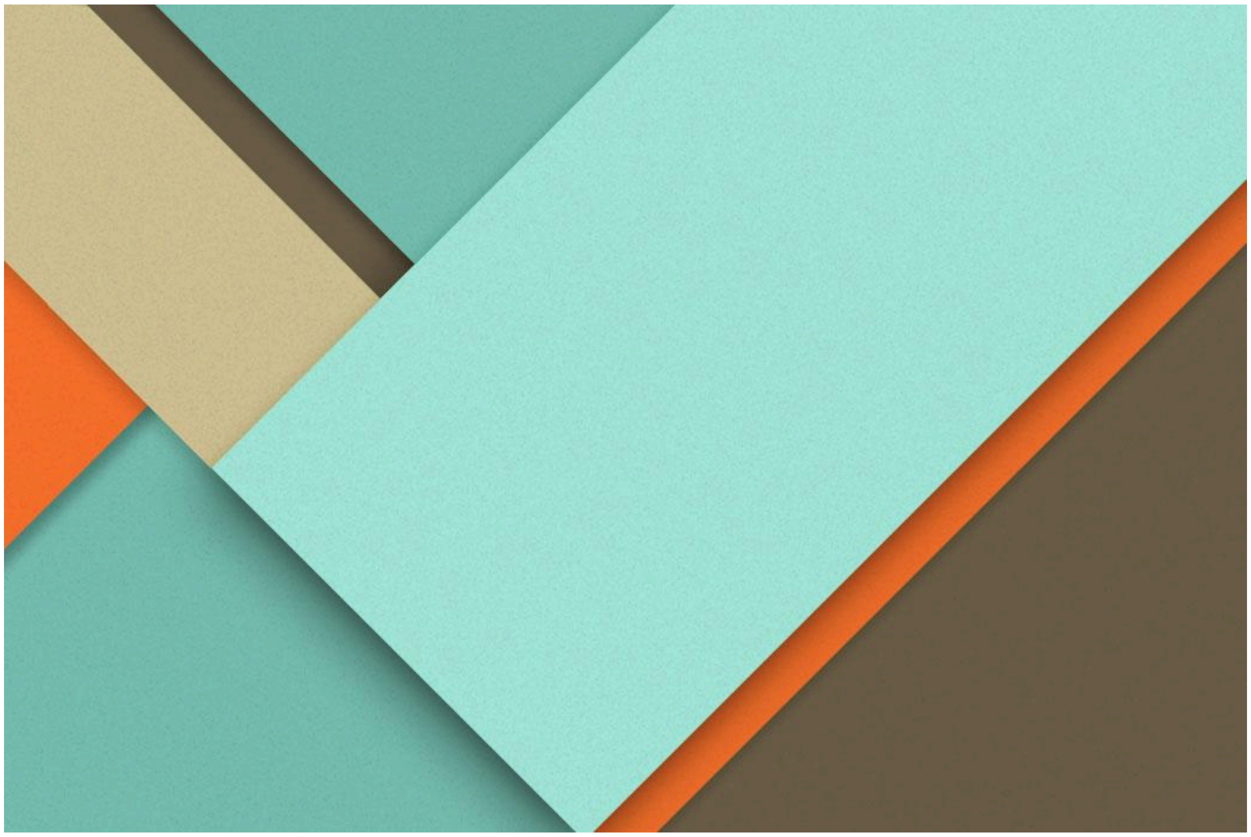

NaukriWallah - A Decentralized Job Marketplace

Team: Bug_Writers

- Anmol Jain (230008009)
- Nidarsana M (230004031)
- Nandini Kumari (230001056)
- Mitanshu Kumawat (230008022)
- Priyanshu Patel (230008026)
- Tripti Anand (230001078)

Part 1: Introduction

This report documents the development of NaukriWallah, a decentralized freelance job marketplace platform built on blockchain. It connects employers and freelancers through a secure, trustless smart contract system, eliminating intermediaries. The platform features escrowed payments, decentralized governance, and dispute resolution mechanisms.

Part 2: Project Objective

- Employers post jobs with escrowed payments.
- Freelancers apply transparently through smart contracts.
- Secure funds during work progress.
- Smart contracts automate payment releases and handle disputes.
- Foster a trustless, decentralized freelance economy.

Part 3: System Architecture

Smart Contract Layer

- Solidity Smart Contract (`JobBoard.sol`)
- Escrowed fund management, job posting, bidding, payout, dispute handling.
- Security via OpenZeppelin's `Ownable`, `ReentrancyGuard`, and gas-efficient data structures.

Application Layer

- Frontend: React.js + Tailwind CSS
- Wallet Connection: MetaMask
- Smart Contract Integration: Ethers.js
- Real-time chat integration using CometChat API.

Service Layer

- Node.js backend (optional for auxiliary services).
- CometChat API for off-chain messaging.

Part 4: Tech Stack

Layer	Technologies
Smart Contracts	Solidity, Hardhat, OpenZeppelin
Frontend	React.js, Tailwind CSS, Ethers.js
Wallet Integration	MetaMask
Communication	CometChat API
Development Tools	GitHub, Hardhat Node

Part 5: Smart Contract Optimization and Security

Optimization Techniques

- Efficient data types (`uint256`) used to minimize gas costs.
- Mappings for quick lookups instead of looping through arrays.
- Minimal on-chain data storage.
- Efficient use of OpenZeppelin's `Counters`.
- Early returns in functions to save computational cost.

Security Mechanisms

- Escrow protection: Funds locked at job creation.
- Access control via `Ownable` and `onlyJobOwner` modifiers.
- Restrict job updates strictly to employers who posted them, ensuring clear separation of access rights between employers and freelancers.
- Reentrancy protection via `ReentrancyGuard`.
- Dispute handling through smart contract functions (`dispute()`, `resolved()`, `revoke()`).
- No personal user data stored on-chain, only Ethereum addresses.

Part 6: Testing Overview

- Unit tests created for all major functionalities.
- Tests cover:
  - Job posting
  - Freelance bidding
  - Payment release post-completion
  - Dispute management workflows
- Executed tests using Hardhat and Chai framework.

Part 7: Practical Problems and Implemented Solutions

Problem	Solution
Unauthorized payment releases	Only job owners can trigger payment release (`onlyJobOwner` modifier).
Double bidding by freelancers	Mapping `hasPlacedBid` prevents duplicate bids.
Freelancers not submitting work	Employers can raise a dispute; admin can revoke job assignment and refund.
Payment security before completion	Escrow logic in the smart contract securely locks the payment until task approval.

Part 8: Future Scope

- Implement a decentralized reputation system (ratings and reviews).
- Integrate DAO-based dispute resolution.
- Support multi-signature wallet controlled job postings.
- Public testnet and mainnet deployment.

Part 9: Conclusion

NaukriWallah leverages blockchain's decentralization to redefine freelance marketplaces by removing intermediaries, ensuring security, and automating workflows through smart contracts. It demonstrates a practical application of decentralized technologies to real-world problems in the gig economy.

Part 10: References and Useful Resources

- Smart Contract Development Frameworks by PatrickAlphaC
- Hardhat Documentation
- OpenZeppelin Contracts Library
- MetaMask Docs - Connecting Dapps
- Ethereum Official Developer Documentation
- Ethers.js Documentation
- CometChat API Documentation