

The background features decorative purple wavy lines in the top-right and bottom-left corners, creating a modern, abstract aesthetic.

# **Blockchain-Based Copyright Protection for Creators**



# PROBLEM STATEMENT

---

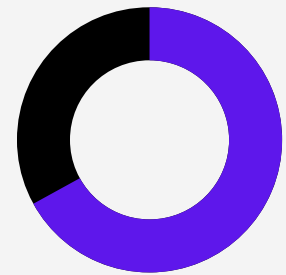
## Challenges Faced by Creators:

- Plagiarism & Unauthorized Use: Lack of a reliable way to prove authorship.
- Centralized Copyright Systems: Expensive, slow, and centralized mechanisms.
- Lack of Transparency: No real-time verification or public proof of ownership.
- Difficulty in Monetization: No direct mechanism for efficient licensing and sales.

## Why This Matters:

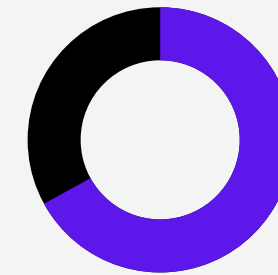
- Creators struggle to protect intellectual property.
- Current systems are inefficient and costly.
- A decentralized, transparent, and immutable solution is needed.

# SOLUTION: BLOCKCHAIN COPYRIGHT PROTECTION SYSTEM



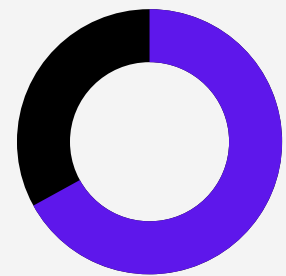
## **Immutable Proof of Ownership**

Register work via hashed file on the blockchain with timestamping.



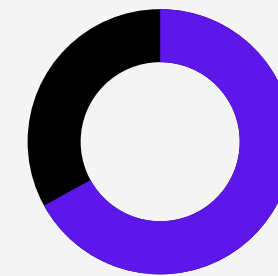
## **IPFS for Secure File Storage**

Decentralized storage with only file hashes on-chain.



## **NFT-Based Copyright Tokens**

Mint NFTs with metadata and IPFS links for royalty tracking.

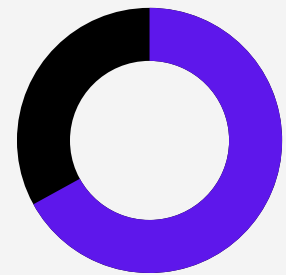


## **Smart Contract-Based Licensing & Monetization**

Automate royalty distribution and enable secure content sales.

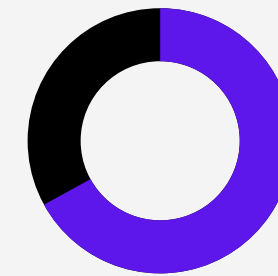
# TECHINICAL ARCHITRCTURE OVERVIEW

## **Blockchain Integration (Algorand)**



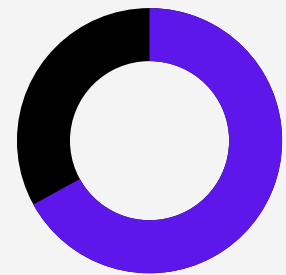
- Leverage Algorand's fast and scalable network for secure and immutable data storage.
- Store file hashes on-chain for proof-of-ownership.

## **Decentralized Storage with IPFS**



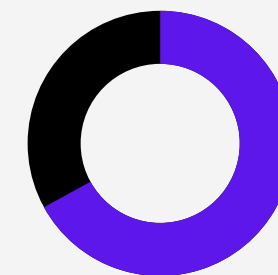
- Store digital content on the InterPlanetary File System (IPFS) for censorship-resistant storage.
- Prevents file tampering while reducing on-chain storage costs.

## **NFT Minting with Algorand Standard Assets (ASA)**



- Mint NFTs to represent digital assets with metadata and ownership details.
- Enable traceability and royalty management.

## **Smart Contract Automation**



- Utilize Algorand smart contracts (PyTeal) for licensing agreements and royalty distribution.
- Automate payments and enforce copyright rules.



# IMPLEMENTATION PLAN & TIMELINE

## *Research & Planning*

- Define system architecture and technical requirements.
- Design smart contract logic and data flow.

## *Development*

- Develop Algorand-based smart contracts using PyTeal.
- Implement NFT minting for copyright registration

## *IPFS Integration & Data Storage*

- Connect IPFS for decentralized file storage.
- Optimize blockchain interactions for efficiency.

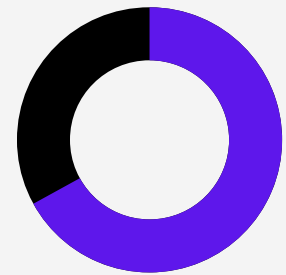
## *Frontend & User Interface*

- Develop a user-friendly UI with React.js & AlgoSDK.
- Implement dashboards for copyright verification & licensing.

## *Testing, Deployment & Launch*

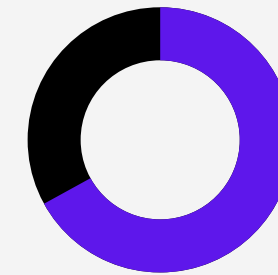
- Conduct security audits and smart contract testing.
- Deploy on Algorand mainnet and onboard initial users.

# EXPECTED IMPACT & OUTCOMES



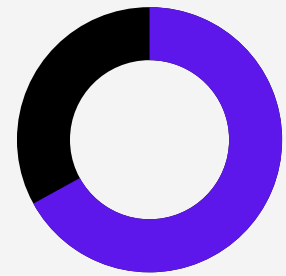
## **Enhanced Copyright Protection**

- Provides creators with immutable proof of ownership using blockchain.
- Reduces plagiarism and unauthorized use of digital content.



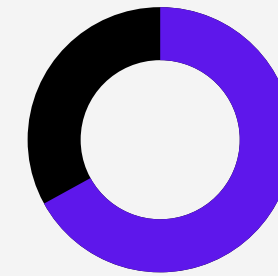
## **Fair Monetization for Creators**

- Smart contracts automate royalty distribution and content licensing.
- Ensures creators receive fair compensation for their work.



## **Decentralized & Transparent System**

- Eliminates reliance on centralized authorities for copyright verification.
- Enables public auditability and real-time verification of ownership.



## **Adoption & Scalability**

- Open-source and hackathon-friendly solution encourages developer contributions.
- Scalable for use across industries like music, art, journalism, and photography.



**Institution:**

**Ajeenkya DY Patil School of Engineering, Pune**



**Team Members:**

◆ **Momin Md. Insha – Engineering, Second Year**

◆ **Parth Ghag – Engineering, First Year**



**Passionate about blockchain and innovative tech solutions.**

# Our Info