Université Euro Méditerranéenne Fès

EuroMed University of Fez

Ecole d'Ingénierie Digitale et d'Intelligence Artificielle (EIDIA)

End-of-Module Project

Program: 2nd-Year Integrated Preparatory Classes

Semestre : 4

Course: Blockchain Technology

Topic:

Proof of work Blockchain

Supervised by:

Pr. TMIMI Mehdi

Prepared by:

ECHCHOURA Mohammed Amine

Academic Year: 2024 / 2025

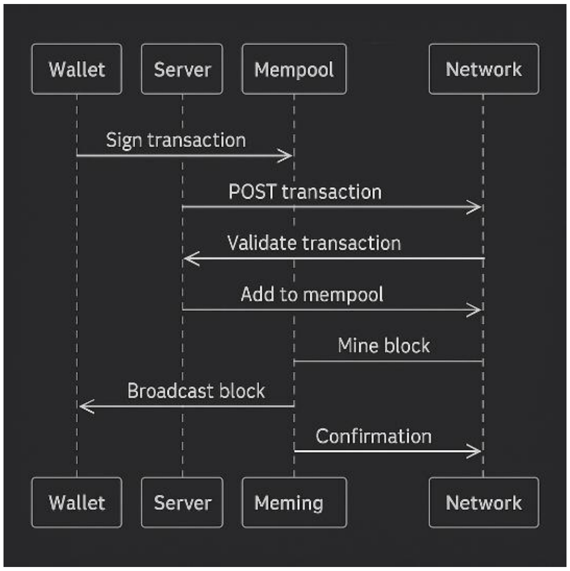
I. Project Overview:

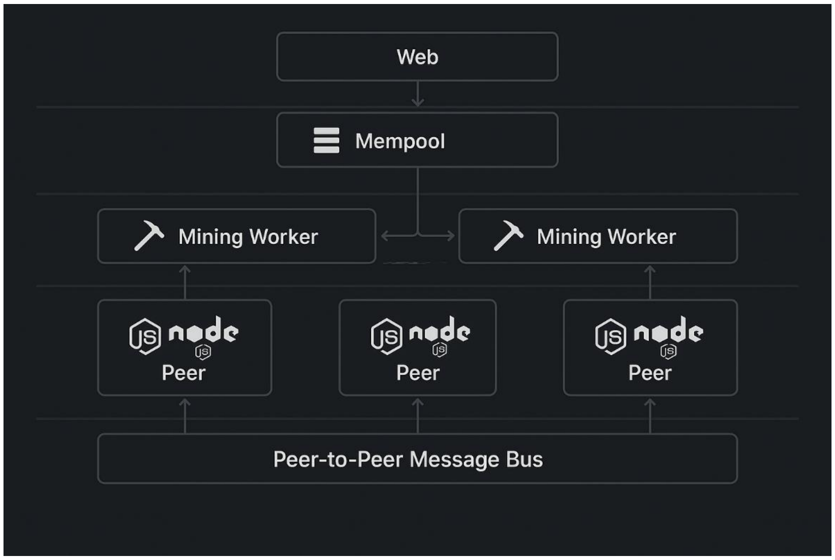
This project implements a **local, single-node proof-of-work (PoW) blockchain** in Node.js, backed by a minimal web UI served with Express. The aim is to give students hands-on exposure to the full life-cycle of a cryptocurrency: wallet creation, transaction signing, mempool management, mining, consensus (difficulty-bound PoW), block persistence, and RESTful APIs for interaction.

II. Goals & Learning Outcomes:

Objective	Hands-on outcome
Understand PoW difficulty	models/block.js hashes until hash.startsWith('0'.repeat(difficulty))
Manage state off-chain	Blocks, mempool, wallets saved as plain JSON files
Verify digital signatures	server.js checks RSA-SHA256 using Node crypto
Build full-stack features	REST API + vanilla JS single-page interface

III. System Architecture:





IV. Key Source Files:

Path	Purpose
models/block.js	Hashing, nonce loop, static `mineNewBlock()`
models/blockchain.js	Validity checks, balance calc, `addBlock()`
models/transaction.js	Plain-object TX with fees & signature
models/wallet.js	RSA-2048 key-pair helper, balance getter
persistence/blockPersistence.js	Save/load individual blocks
persistence/mempoolPersistence.js	Read/write `mempool.json`
persistence/walletPersistence.js	Cache wallet balances
server.js	All REST endpoints, static file serving, optional auto-miner
web/index.html	Single-page UI with embedded CSS & JS

V. API Reference:

Method	Endpoint	Body / Param	Action
GET	/blocks	–	Return full blockchain
GET	/wallets	–	Wallet balances snapshot
GET	/mempool	–	Pending TX list
POST	/transactions/new	{sender,recipient,amount,fee,signature}	Validate & queue TX
POST	/mine	{minerAddress}	Assemble block from mempool and run PoW
GET	/keys/:wallet/private	–	(Dev) download PEM key

VI. Consensus & Mining Logic:

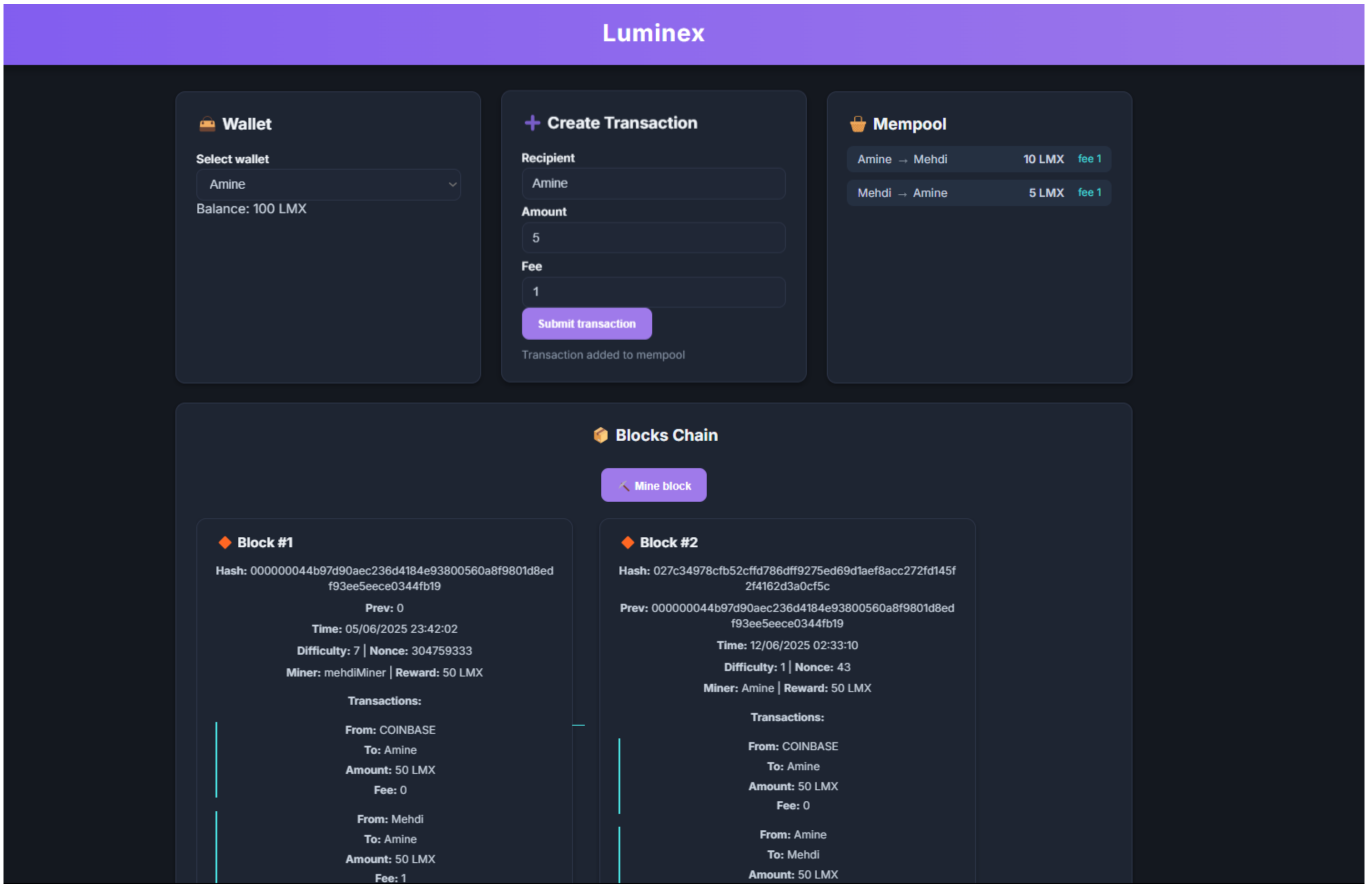
1. **Difficulty** is a simple integer constant (difficulty = 4 by default)
2. A mining request bundles:
   - Coinbase TX (50 uemfCoin).
   - All mempool TXs, prioritised by fee (highest first).
3. **Block.mineNewBlock()** increments nonce until the SHA-256 header hash meets the target.
4. On success the block is:
   - Validated by Blockchain.addBlock() (hash chain, signatures, balances).
   - Written to /blocks/HEIGHT.json.
   - Confirmed TXs are purged from mempool.json.

VII. Persistence Strategy:

Store	**Location**	**Structure**
Blocks	blocks/0.json … N.json	Full block objects
Mempool	database/mempool.json	[tx, …]
Wallets	database/wallets.json	{ address: balance }

VIII. Front-End Highlights:

- **Chain view** – horizontal cards with widened background to avoid “white zipper” artefact.
- **Mempool list** – flex column styled like a transaction queue.
- **Stats ribbon** – centred counts (height, mempool size, difficulty).
- **Dark theme** – softer slate tones (#1e293b) plus accent teal for buttons.



Luminex

Wallet

Select wallet

Amine

Balance: 100 LMX

Create Transaction

Recipient

Amine

Amount

5

Fee

1

Transaction added to mempool

Mempool

Amine → Mehdi	10 LMX	fee 1
Mehdi → Amine	5 LMX	fee 1

Blocks Chain

Block #1

Hash: 000000044b97d90aec236d4184e93800560a8f9801d8ed f93ee5eece0344fb19

Prev: 0

Time: 05/06/2025 23:42:02

Difficulty: 7 | **Nonce:** 304759333

Miner: mehdiMiner | **Reward:** 50 LMX

Transactions:

From: COINBASE
To: Amine
Amount: 50 LMX
Fee: 0

From: Mehdi
To: Amine
Amount: 50 LMX
Fee: 1

Block #2

Hash: 027c34978cfb52cffd786dff9275ed69d1aef8acc272fd145f 2f4162d3a0cf5c

Prev: 000000044b97d90aec236d4184e93800560a8f9801d8ed f93ee5eece0344fb19

Time: 12/06/2025 02:33:10

Difficulty: 1 | **Nonce:** 43

Miner: Amine | **Reward:** 50 LMX

Transactions:

From: COINBASE
To: Amine
Amount: 50 LMX
Fee: 0

From: Amine
To: Mehdi
Amount: 50 LMX

IX. Test Checklist:

Scenario	Expectation
Invalid RSA signature	400 "Invalid signature"
Double-spend attempt	TX rejected, mempool unchanged
Mining with empty mempool	Block with coinbase only
Restart server	Chain + mempool reload correctly
Editing difficulty, reward in code	New blocks honour settings

X. Conclusion:

The final iteration streamlines persistence, cleans up the UI, and isolates concerns into four clear layers (UI → Server → Persistence → Models). That makes the code base easier to extend (e.g., adding peer-to-peer networking or swapping PoW for PoS) and more reliable for grading and demonstrations.