# MasumiRanker — AI Agent Rating and Discovery Platform

**Hackathon Project Specification**

Hackathon Duration: 24 hours  
Team Members:

* Member 1 – Frontend Developer (React) **Anastasia He**
* Member 2 – Backend Developer (Node.js / Python) **Thomas Wang**
* Member 3 – ML Specialist (PyTorch) /Database **Claire Wen**

## Project Objective

MasumiRanker is a decentralized web platform that allows users to:

* Explore AI Agents built on the Masumi Network
* Submit ratings and reviews
* Browse agents by category and ranking
* View audit-friendly metadata (timestamp + hash) for each rating

**All ratings are timestamped and cryptographically hashed to ensure transparency and integrity.**

Since the official Masumi API and dataset are limited, we will simulate a full backend and data model to support functionality and testing.

## Key Features

### 1. Home Page: Agent Listing

* Display all agents with summary information (name, category, average score, number of ratings)
* Filter agents by category
* Sort by average rating

### 2. Agent Detail Page

* Full details: name, description, category, DID, and external link
* User ratings and comments
* Submit new ratings (score 1–5 + comment)
* Display timestamp and hashed record (simulated on-chain log)

### 3. Ranking Page

* Show top-rated agents per category
* Allow filtering and sorting by rating or usage
* Clickable links to detail pages

## Team Responsibilities

| **Member** | **Role** | **Responsibilities** |
| --- | --- | --- |
| 1 | Frontend | React pages, UI components, star rating component, API integration |
| 2 | Backend | Build REST API, serve mock data, implement simulated audit log |
| 3 | ML/AI/Database | Optional: comment sentiment analysis, hot keyword extraction, basic recommendation logic |
| PM | Planning | Write documentation, define APIs, coordinate demo and pitch materials |

## API Specification Overview

Defined in document/api\_spec.md

| **Endpoint** | **Method** | **Description** |
| --- | --- | --- |
| /agents | GET | List all agents |
| /agents/:id | GET | Get full detail of one agent |
| /ratings | POST | Submit a rating |
| /ratings/:agent\_id | GET | Fetch all ratings for one agent |
| /rankings | GET | Get top-rated agents by category |

## Simulated Database Schema

Defined in document/data\_schema.md

### agents\_mock.json

{

"id": "agent001",

"name": "Coding Mentor",

"category": "Education",

"description": "Helps you learn to code.",

"did": "did:masumi:abc123",

"url": "https://example.com/codingmentor",

"avg\_score": 4.6,

"num\_ratings": 12

}