On-Campus Task Platform

The **On-Campus Task Platform** is designed to help students fulfill tasks by allowing them to post requests for items or services they need, whether on or around campus.

By group _6

Why This Platform is Needed

Problem

- Students often face challenges in obtaining items due to time constraints, lack of nearby stores, or transportation issues.

Solution

- The platform centralizes task requests, making it easier for students to post what they need and find others willing to help.It incorporates a **dual rating system**, ensuring transparency and trust.

Key Features

1. **Request Creation**

   Students can post detailed requests for items or services they need, including the location and reward.

2. **Task Browsing and Application**

   Users can browse tasks and apply to complete them, offering their own proposal and timeline.

3. **Dual Rating System**

   Requesters and task doers can rate each other, building trust and transparency.

4. **Joint Tasks**

   Multiple students can create a shared request, splitting the cost and reward to make the task more affordable and rewarding.

5. **Real-Time Chat**

   Integrated chat for easy communication between requesters and task doers.

How It Works

1. A Requester creates a task with details, including what is needed, where to get it, and the reward.
2. Other students can browse available tasks and apply with their proposals.
3. The Requester reviews applications, negotiates if needed, and selects the best candidate.
4. Once the task is completed, both the requester and task doer **rate each other** based on their experience, ensuring trust and transparency.

Technical Requirements

Frontend Development

- React.js will be used for the web application, ensuring a responsive and user-friendly interface.

Backend Development

- The backend will be built using Node.js or Django to manage server-side operations, such as handling user requests, task management, and communication between users.

Database

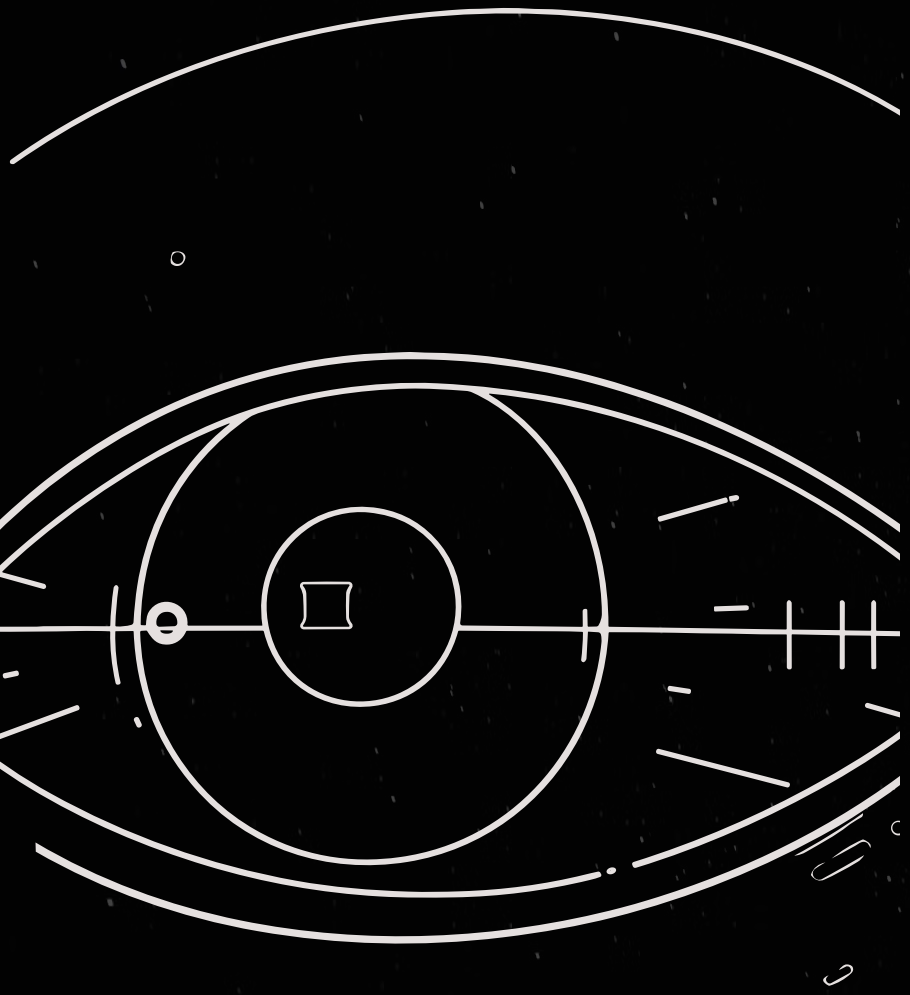
- MongoDB will store user data, tasks, ratings, and transaction information securely, ensuring efficient data retrieval and management.

Real-Time Features

- Firebase will be used for real-time chat and notifications, keeping users informed about task status and updates.

Hosting

- The platform will be hosted on Netlify or Google Cloud, ensuring scalability, security, and reliability.



Future Improvements

Enhanced Search

Allowing users to find tasks based on specific criteria like location , task type, deadline, or reward.

Task Progress Tracking

Allowing requesters and task doers to update the status of the task

Geo-Location Integration

A**utomatically detect user locations** and provide task recommendations based on proximity.

Achievements

Introduce **elements** such as badges, leaderboards, and achievements to motivate users.