# Reddit Clone Implementation Report

## Project Overview

This report details the implementation of a Reddit clone system utilizing a REST API architecture for client-server communication. The project implements core Reddit functionality including user registration, forum management, posting, commenting, voting, and direct messaging.

## Team Members

1. Arpita Patnaik
2. Bala Tripura Kumari Bodapati

## Demo Link

https://www.youtube.com/watch?v=RSbL\_fuPvZ8&feature=youtu.be

## System Architecture

### Components

#### REST API Server (server.go)

* Handles HTTP requests using the Gorilla Mux router
* Implements CORS and logging middleware
* Routes requests to appropriate engine handlers
* Response standardization using common Response struct

#### Social Engine (social\_engine.go)

* Core business logic implementation
* Actor-based concurrency using Proto Actor framework
* In-memory data structures for users, forums, posts, and messages
* Thread-safe operations using mutex locks

#### REST Client (rest\_client.go)

* Command-line interface for interacting with the server
* Supports all core Reddit functionalities
* Handles JSON serialization/deserialization
* Configurable timeout settings

#### Protocol Definitions (messages.proto)

* Protocol Buffer definitions for all message types
* Structured data models for content, feedback, and chat
* Clear separation between request and response messages

### Key Design Patterns

#### Actor Model

* Uses Proto Actor for concurrent message handling
* Ensures thread safety for shared state
* Enables scalable message processing

#### Repository Pattern

* Centralized data management in SocialEngine
* Clean separation of concerns
* Consistent data access patterns

## Core Functionality Implementation

### User Management

func (s \*Server) registerUser(w http.ResponseWriter, r \*http.Request) {  
 // User registration implementation  
}

* Simple username-based registration
* Unique username validation
* Activity status tracking

### Forum Operations

* Forum creation with name and description
* Join/leave forum functionality
* Member tracking and forum statistics
* Forum content feed management

### Content Management

* Post creation with title and body
* Hierarchical comment system
* Content ID generation for unique identification
* Support for reposting/sharing content

### Voting System

* Upvote/downvote functionality
* Points calculation
* User karma tracking
* Hot score computation for content ranking

### Direct Messaging

* User-to-user messaging
* Message status tracking (seen/unseen)
* Message retrieval with pagination
* Real-time message delivery

## Running Multiple Clients

### Setup Instructions

1. Start the server:

go run cmd/server/main.go -port 8080 -actor-port 8085

2. Start multiple client instances:

go run client/rest\_client.go

## Testing Multiple Client Interactions

1. User Registration

A computer screen shot of white text

Description automatically generated

A computer screen shot of white text

Description automatically generated

1. Forum Creation and Joining



A black background with white text

Description automatically generated

1. Create Post



1. Get Feed

A screenshot of a computer

Description automatically generated

1. Exit

A screen shot of a computer

Description automatically generated

## Conclusion

The implemented Reddit clone successfully demonstrates core functionality with a clean separation between client and server components. The REST API provides a standardized interface for client-server communication, while the actor-based engine ensures thread-safe operations. The system supports multiple simultaneous clients and implements all required features including user management, content creation, voting, and direct messaging.