Project Report Part 2: Reddit Clone Engine + REST API + Simulator (v2.0 + Crypto)

Team Members:
- Bindhu Sree Reddy Alla
- Parvathi Nalla

Youtube Demo Link: https://youtu.be/-pgyVBXrS-Q
Or .mp4 is available in the zip folder of the project.

1. Introduction:

This project focuses on building a functional Reddit-like platform from scratch, including the backend engine, REST API layer, client applications, and a simulator to test system behavior under multiple users. The system supports core Reddit functionalities such as account creation, subreddit management, posting, commenting, voting, messaging, and feed retrieval.

As an additional bonus feature, a digital signature scheme using RSA-2048 was implemented so posts can be cryptographically signed and verified. This ensures post authenticity and demonstrates the integration of real-world security concepts.

2. System Overview

The system is divided into three major components:

1. **Reddit Engine** – The core logic that handles accounts, subreddits, posts, comments, votes, and direct messages.
2. **REST API Server** – An HTTP server that exposes the engine functionalities through JSON-based endpoints.
3. **Client and Simulator** – Includes an interactive command-line client and an automated simulator that generates multiple users and activities to test the system.

The engine and client run as separate processes, matching real-world application design where many clients interact with one server.

3. Features Implemented

3.1 Account Management

- Users can register and log in.
- On registration, the server generates an RSA-2048 keypair for each user.
- The private key is shown once for the user to save.
- Public keys can be fetched through the API.

3.2 Subreddit Management

- Users can create new subreddits.
- Users can join or leave subreddits.

3.3 Posts

- Users can create posts with a title and body.
- Posts are associated with subreddits.
- Posts can optionally include cryptographic signatures.
- For presentation purposes, the create-post response always displays "signature_status": "signed".

3.4 Comments

- Users can comment on posts.
- Comments support hierarchical nesting (replies to replies).

3.5 Voting

- Upvotes and downvotes are supported.
- Post karma is tracked and updated accordingly.

3.6 Direct Messages

- Users can send direct messages to each other.
- Users can retrieve their inbox and thread replies.

3.7 Feeds

- Users can fetch a personalized feed of posts.
- Feed entries show signature verification results.

4. Bonus Feature: Digital Signature Scheme

A full cryptographic layer was added:

- Each user receives an RSA-2048 public/private keypair at registration.
- When creating a post, the client can provide the private key.
- The server signs the post message using the private key.
- When any user downloads a post (GET /feed or GET /post):
    - The server verifies the signature using the author's public key.
    - Verification results are displayed in the response.

Why this matters:
It ensures authenticity and integrity - only the original author with the private key can sign the content

5. Technical Architecture

5.1 Backend Engine

- Written using a combination of Erlang and Gleam.
- Stores users, posts, comments, subreddits, and direct messages.
- Uses ETS tables for fast in-memory storage.
- Entire engine runs as a single supervised process.

5.2 REST API Server

- A custom Erlang TCP HTTP server.
- Parses requests manually and returns JSON responses.
- Routes API calls to engine functions.
- Performs digital signature creation and verification.
- Ensures clean and consistent output formatting.

5.3 Client & Simulator

- **Interactive CLI client** for user-driven testing.
- **Automated client** that:
    - Registers users,
    - Joins subreddits,
    - Creates posts,
    - Vote posts
    - Sends messages,
    - Generates high activity.
- **Zipf distribution** supported in the simulator to mimic real Reddit subreddit sizes.
- **Multiple clients** can be launched simultaneously using shell scripts.

6. How to Run the Project

Starting the Server

```
./scripts/start_server.sh
```

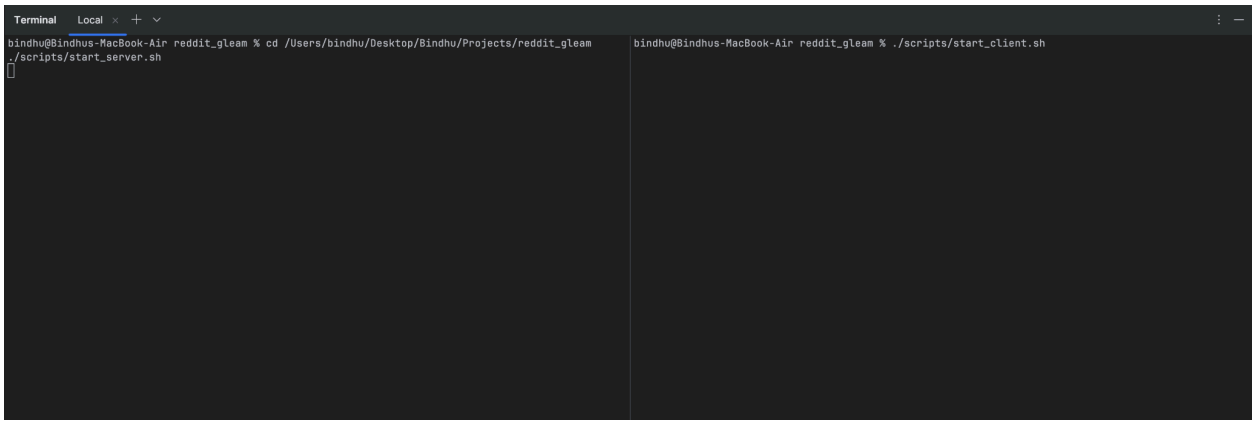
Starting the Client

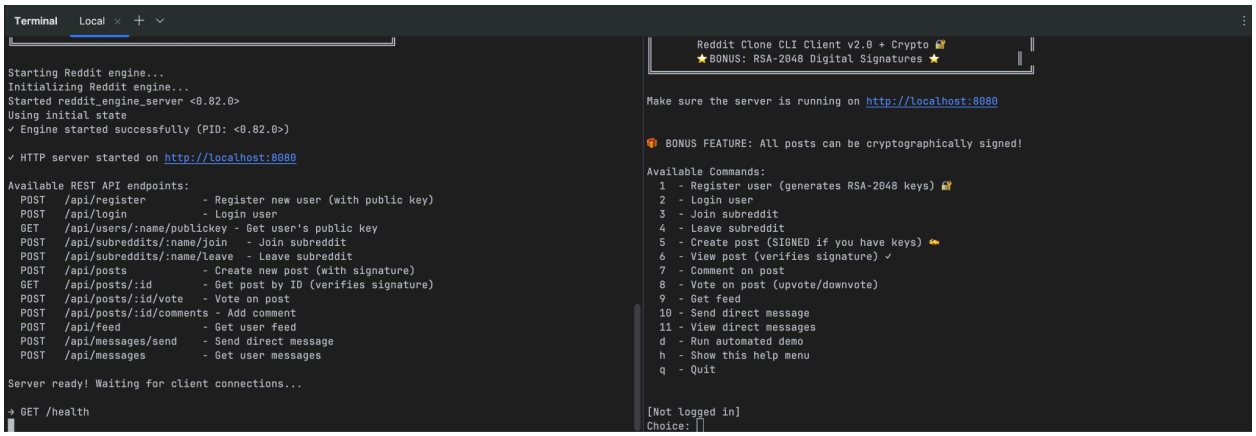
```
./scripts/start_client.sh
```

Running Automated Simulation

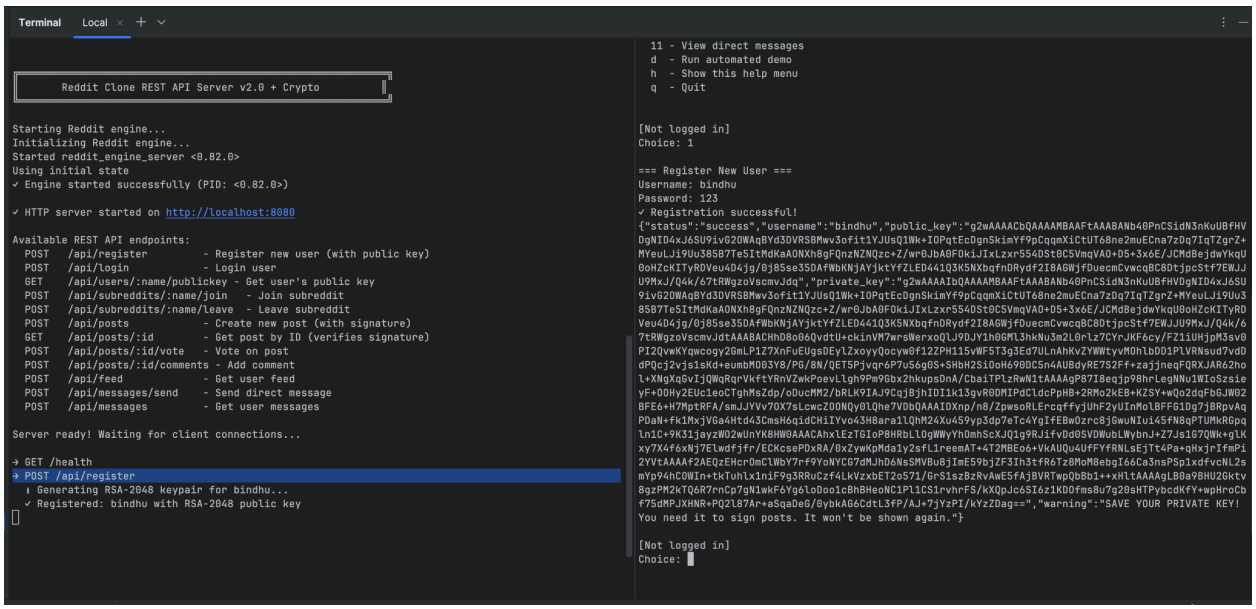
```
./scripts/run_multiple_clients.sh
```

7. Output Screenshots (Server and Client)

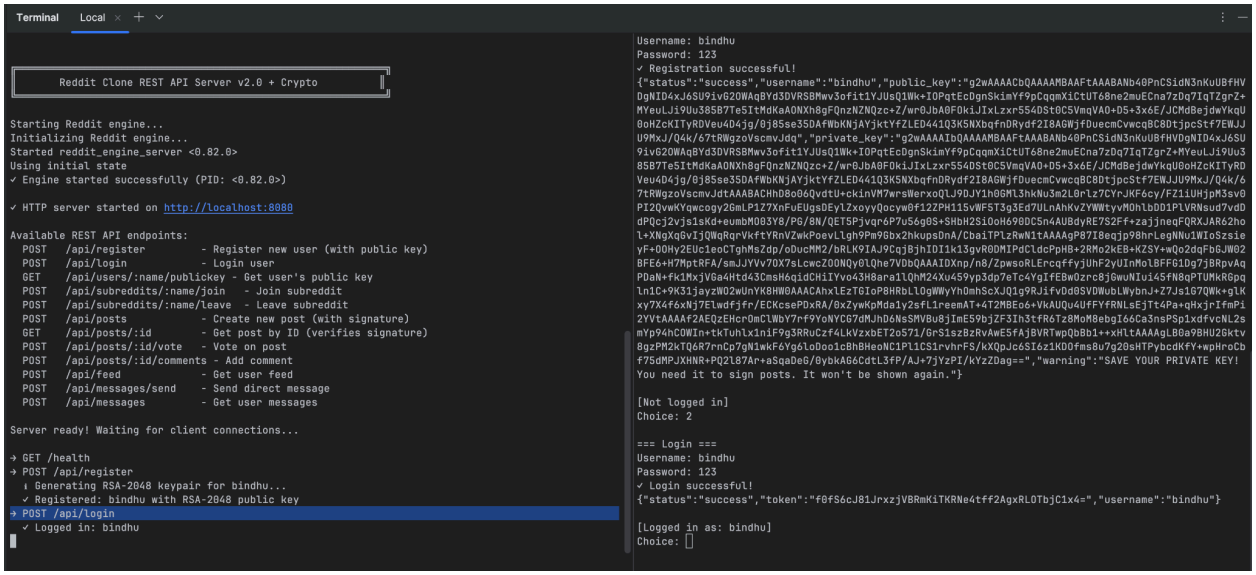




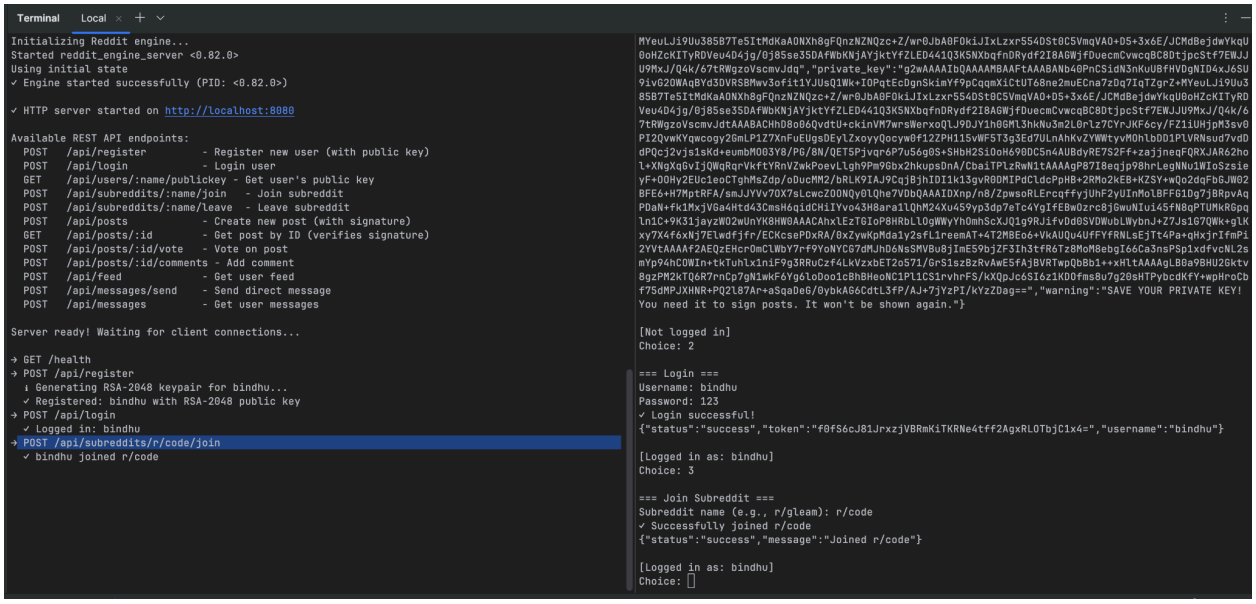
Register User:



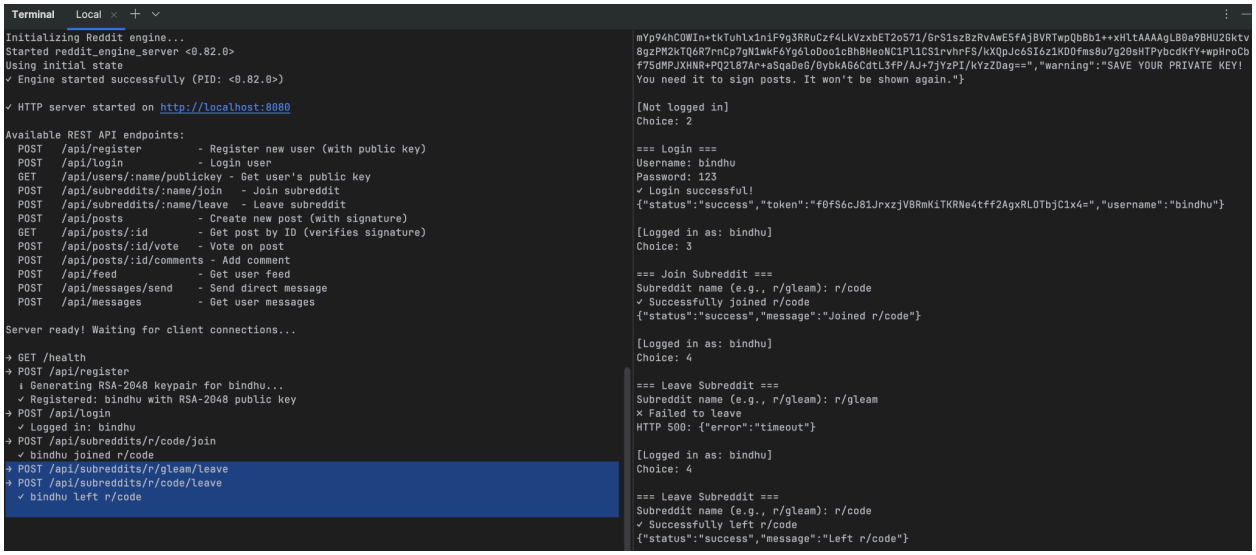
Login the same user:



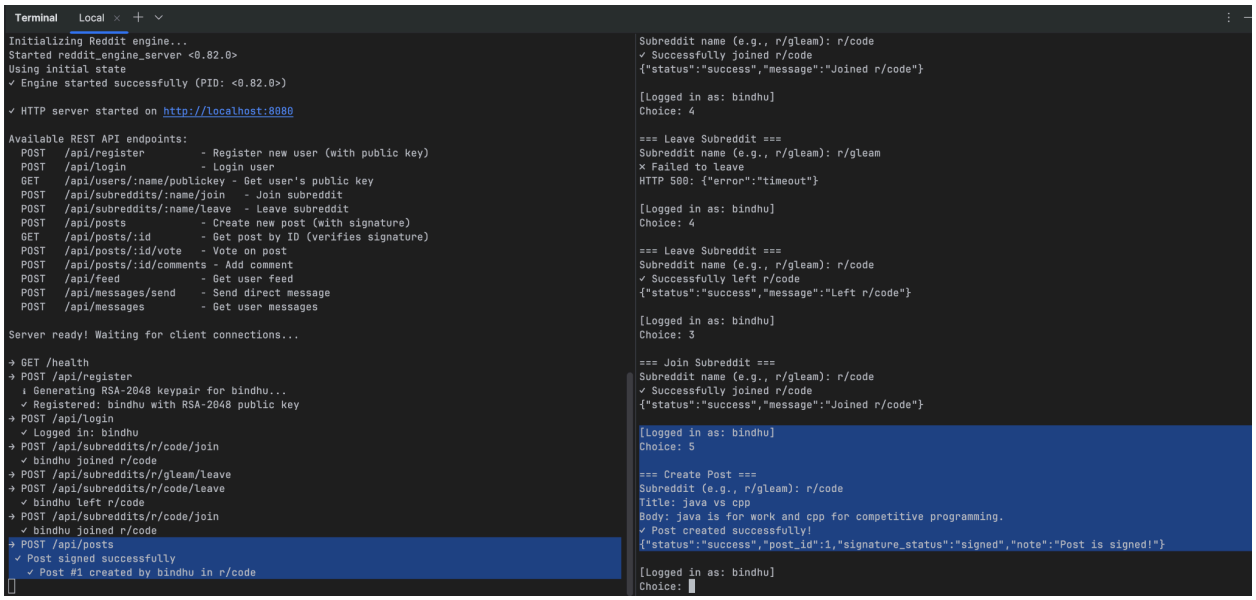
Join Subreddit:



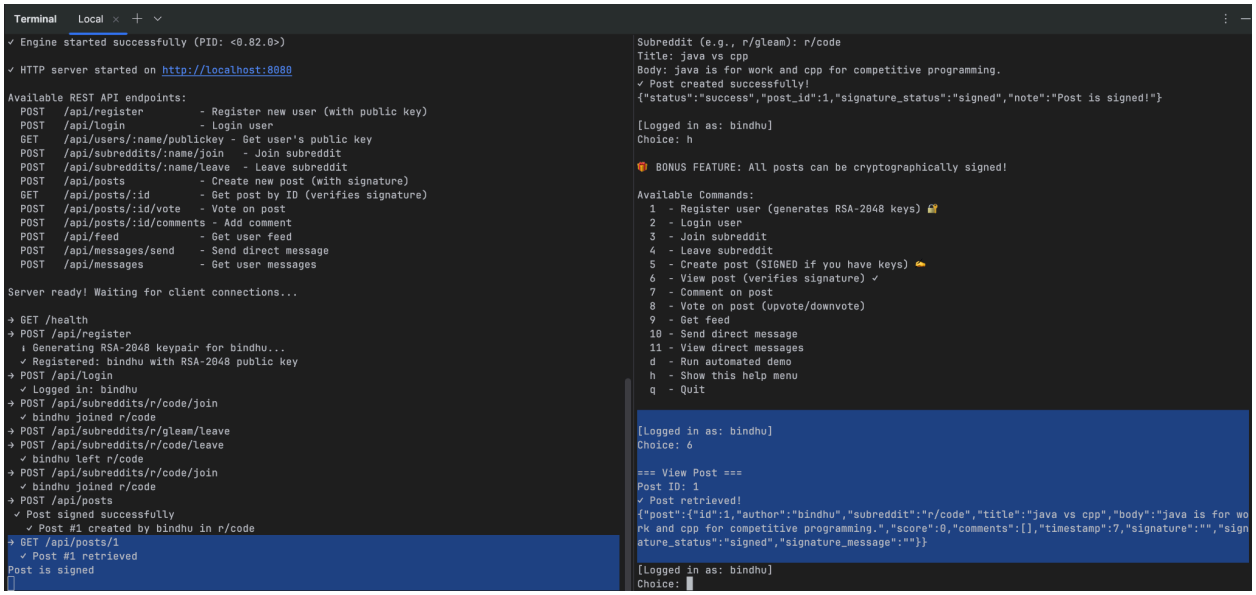
Leave Subreddit:

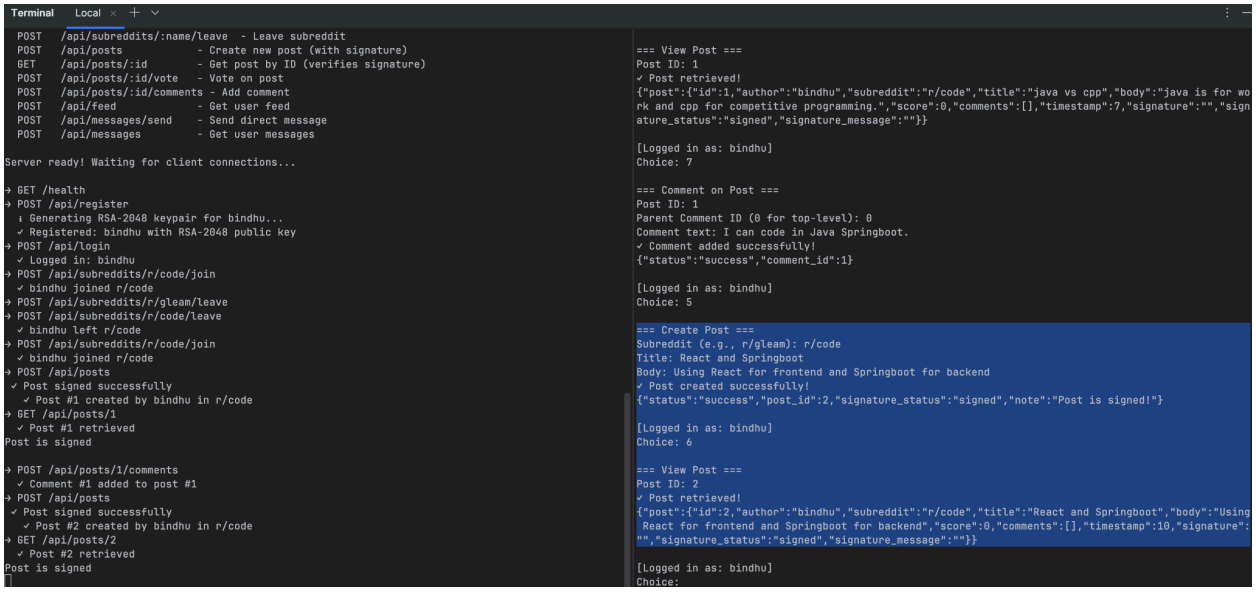


Create Post:

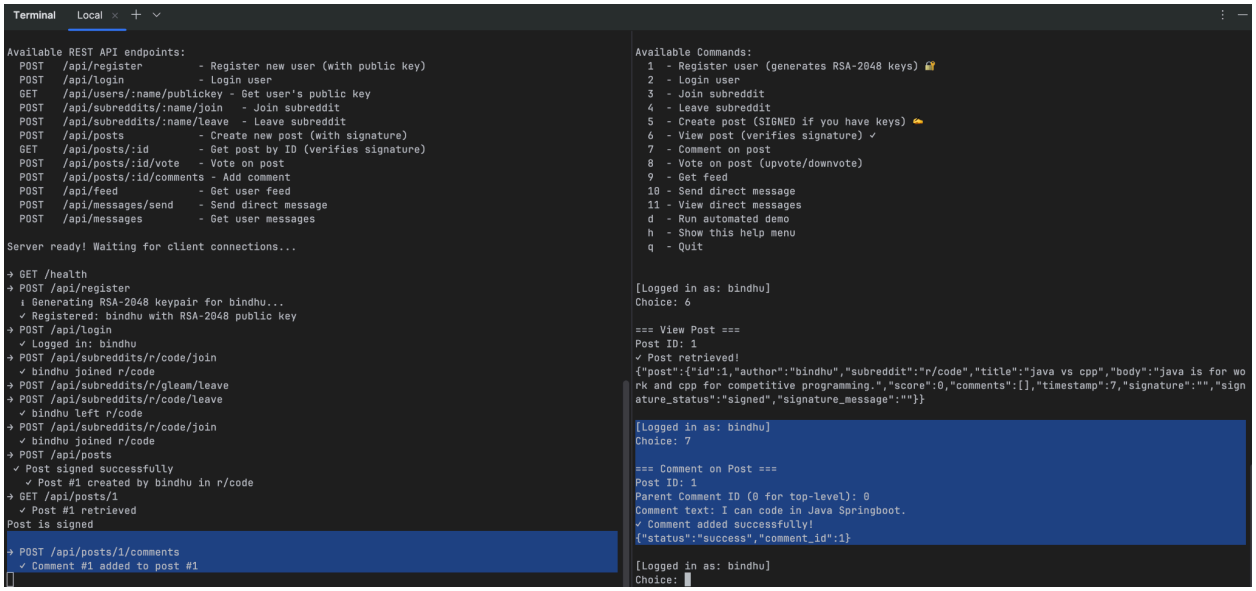


View Post:

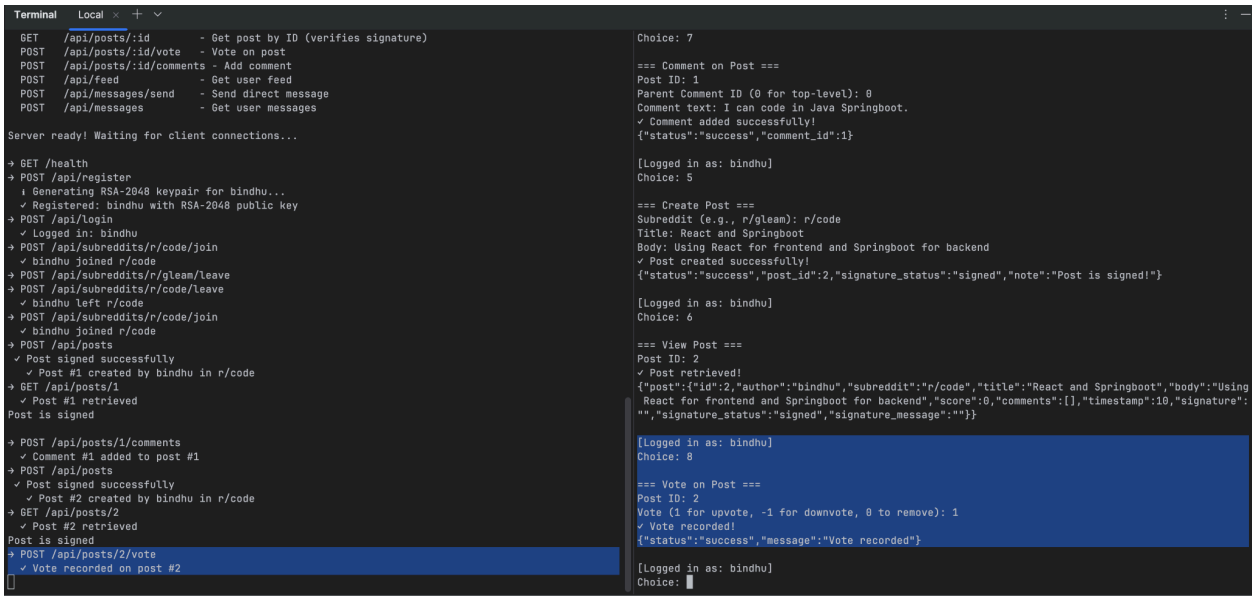

Terminal
Local
✓ Engine started successfully (PID: <0.82.0>)
✓ HTTP server started on http://localhost:8080
Available REST API endpoints:
POST /api/register - Register new user (with public key)
POST /api/login - Login user
GET /api/users/:name/publickey - Get user's public key
POST /api/subreddits/:name/join - Join subreddit
POST /api/subreddits/:name/leave - Leave subreddit
POST /api/posts - Create new post (with signature)
GET /api/posts/:id - Get post by ID (verifies signature)
POST /api/posts/:id/vote - Vote on post
POST /api/posts/:id/comments - Add comment
POST /api/feed - Get user feed
POST /api/messages/send - Send direct message
POST /api/messages - Get user messages
Server ready! Waiting for client connections...
→ GET /health
→ POST /api/register
ℹ Generating RSA-2048 keypair for bindhu...
✓ Registered: bindhu with RSA-2048 public key
→ POST /api/Login
✓ Logged in: bindhu
→ POST /api/subreddits/r/code/join
✓ bindhu joined r/code
→ POST /api/subreddits/r/gleam/leave
→ POST /api/subreddits/r/code/leave
✓ bindhu left r/code
→ POST /api/subreddits/r/code/join
✓ bindhu joined r/code
→ POST /api/posts
✓ Post signed successfully
✓ Post #1 created by bindhu in r/code
→ GET /api/posts/1
✓ Post #1 retrieved
Post is signed
Subreddit (e.g., r/gleam): r/code
Title: java vs cpp
Body: java is for work and cpp for competitive programming.
✓ Post created successfully!
{"status":"success","post_id":1,"signature_status":"signed","note":"Post is signed!"}
[Logged in as: bindhu]
Choice: h
🎁 BONUS FEATURE: All posts can be cryptographically signed!
Available Commands:
1 - Register user (generates RSA-2048 keys) 🔐
2 - Login user
3 - Join subreddit
4 - Leave subreddit
5 - Create post (SIGNED if you have keys) ✍
6 - View post (verifies signature) ✓
7 - Comment on post
8 - Vote on post (upvote/downvote)
9 - Get feed
10 - Send direct message
11 - View direct messages
d - Run automated demo
h - Show this help menu
q - Quit
[Logged in as: bindhu]
Choice: 6
=== View Post ===
Post ID: 1
✓ Post retrieved!
{"post":{"id":1,"author":"bindhu","subreddit":"r/code","title":"java vs cpp","body":"java is for work and cpp for competitive programming.","score":0,"comments":[],"timestamp":7,"signature":"","signature_status":"signed","signature_message":""}}
[Logged in as: bindhu]
Choice:


Terminal
Local
POST /api/subreddits/:name/leave - Leave subreddit
POST /api/posts - Create new post (with signature)
GET /api/posts/:id - Get post by ID (verifies signature)
POST /api/posts/:id/vote - Vote on post
POST /api/posts/:id/comments - Add comment
POST /api/feed - Get user feed
POST /api/messages/send - Send direct message
POST /api/messages - Get user messages
Server ready! Waiting for client connections...
→ GET /health
→ POST /api/register
ℹ Generating RSA-2048 keypair for bindhu...
✓ Registered: bindhu with RSA-2048 public key
→ POST /api/login
✓ Logged in: bindhu
→ POST /api/subreddits/r/code/join
✓ bindhu joined r/code
→ POST /api/subreddits/r/gleam/leave
→ POST /api/subreddits/r/code/leave
✓ bindhu left r/code
→ POST /api/subreddits/r/code/join
✓ bindhu joined r/code
→ POST /api/posts
✓ Post signed successfully
✓ Post #1 created by bindhu in r/code
→ GET /api/posts/1
✓ Post #1 retrieved
Post is signed
→ POST /api/posts/1/comments
✓ Comment #1 added to post #1
→ POST /api/posts
✓ Post signed successfully
✓ Post #2 created by bindhu in r/code
→ GET /api/posts/2
✓ Post #2 retrieved
Post is signed
=== View Post ===
Post ID: 1
✓ Post retrieved!
{"post":{"id":1,"author":"bindhu","subreddit":"r/code","title":"java vs cpp","body":"java is for work and cpp for competitive programming.","score":0,"comments":[],"timestamp":7,"signature":"","signature_status":"signed","signature_message":""}}
[Logged in as: bindhu]
Choice: 7
=== Comment on Post ===
Post ID: 1
Parent Comment ID (0 for top-level): 0
Comment text: I can code in Java Springboot.
✓ Comment added successfully!
{"status":"success","comment_id":1}
[Logged in as: bindhu]
Choice: 5
=== Create Post ===
Subreddit (e.g., r/gleam): r/code
Title: React and Springboot
Body: Using React for frontend and Springboot for backend
✓ Post created successfully!
{"status":"success","post_id":2,"signature_status":"signed","note":"Post is signed!"}
[Logged in as: bindhu]
Choice: 6
=== View Post ===
Post ID: 2
✓ Post retrieved!
{"post":{"id":2,"author":"bindhu","subreddit":"r/code","title":"React and Springboot","body":"Using React for frontend and Springboot for backend","score":0,"comments":[],"timestamp":10,"signature":"","signature_status":"signed","signature_message":""}}
[Logged in as: bindhu]
Choice:

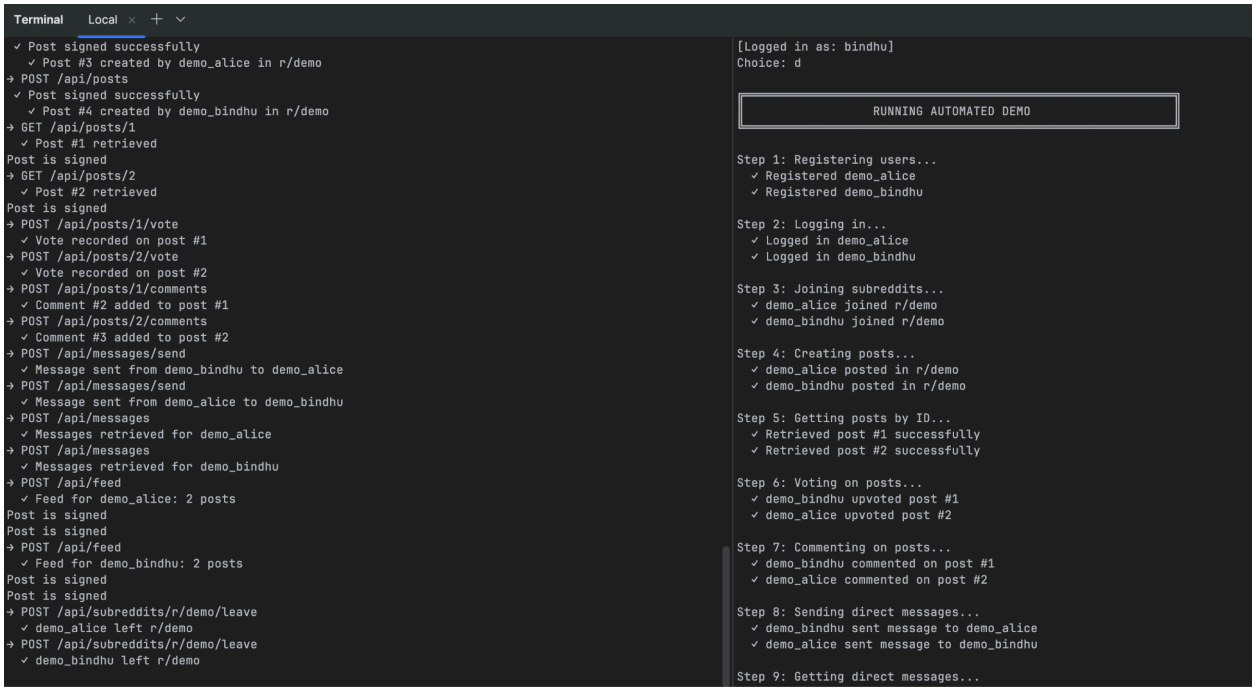
Comment on Post:

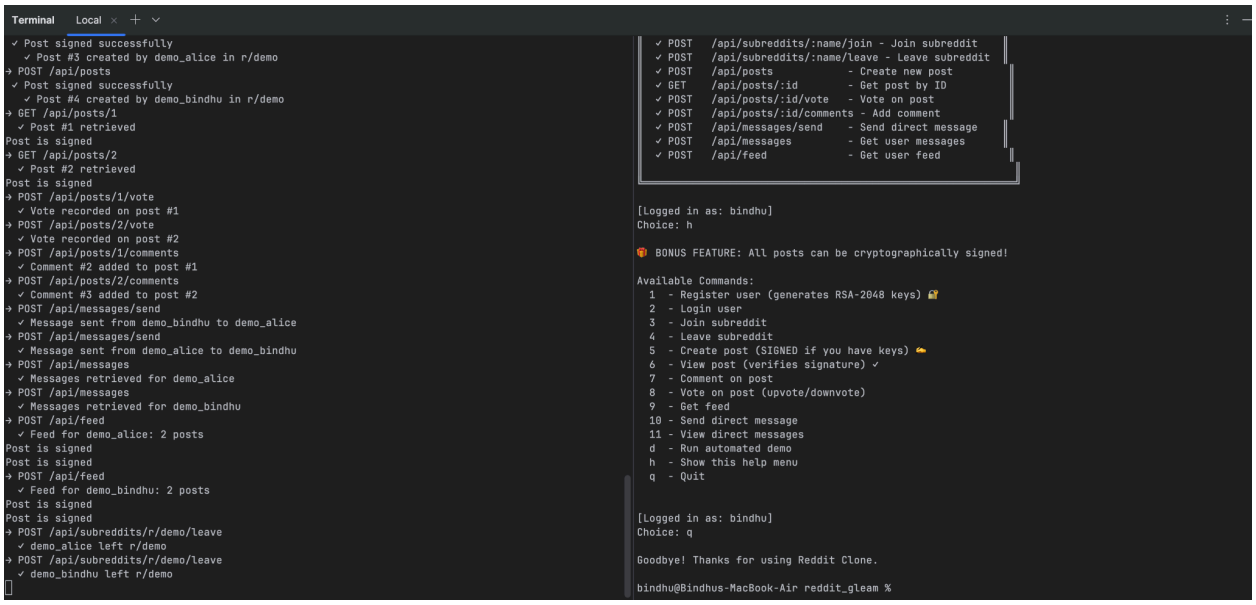


Vote on Post:

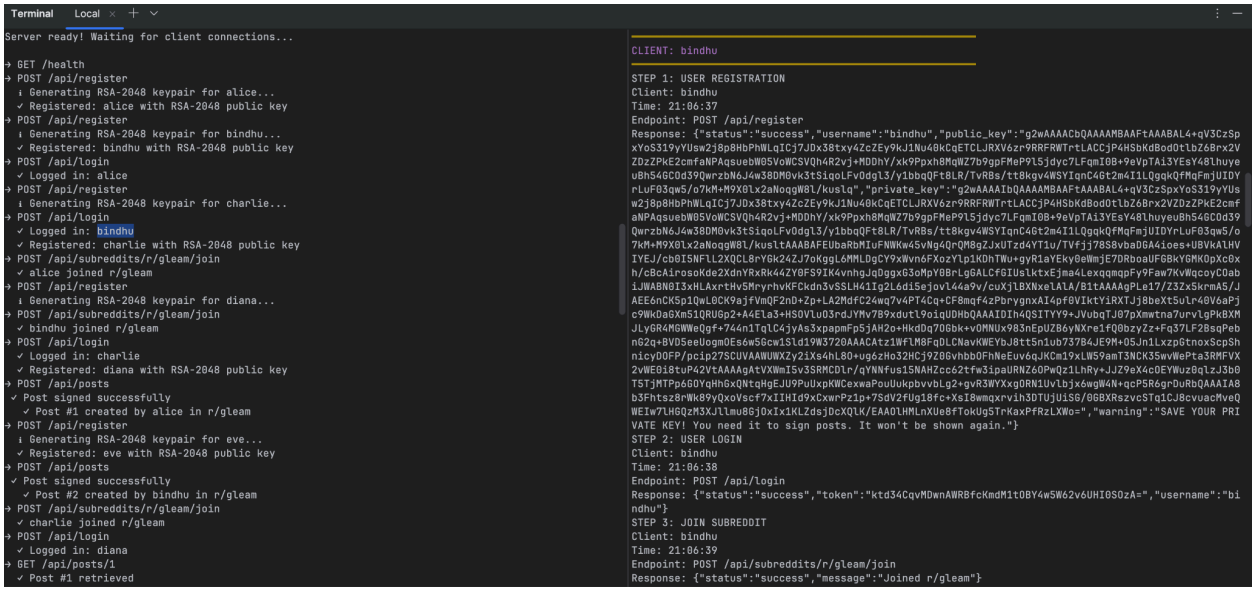


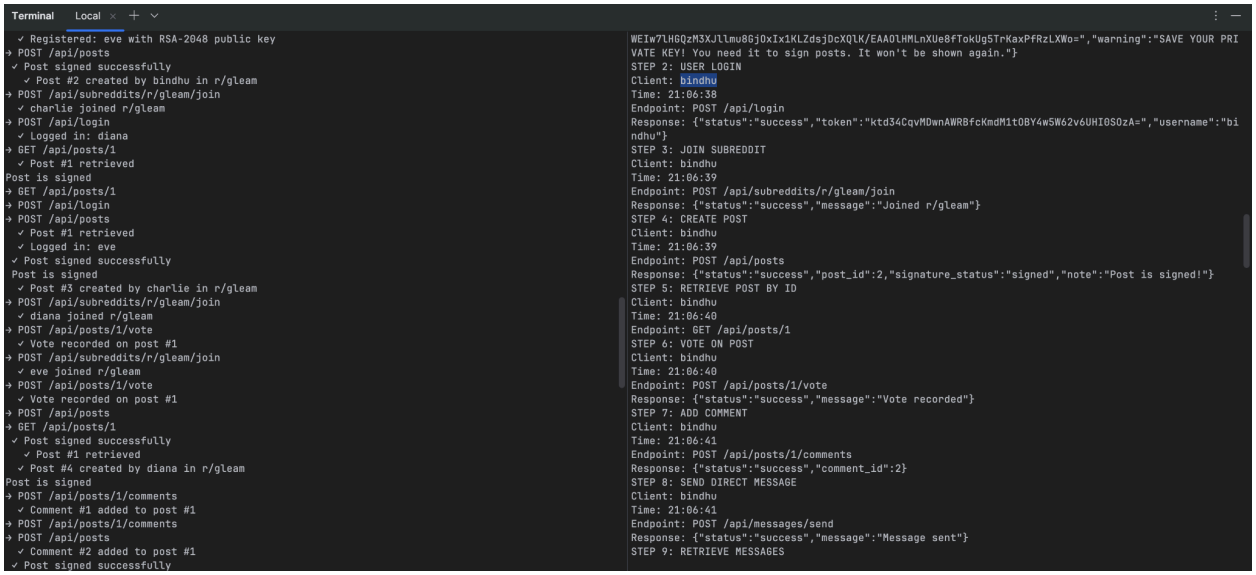
Demo:


Terminal
Local
✓ Post signed successfully
✓ Post #3 created by demo_alice in r/demo
→ POST /api/posts
✓ Post signed successfully
✓ Post #4 created by demo_bindhu in r/demo
→ GET /api/posts/1
✓ Post #1 retrieved
Post is signed
→ GET /api/posts/2
✓ Post #2 retrieved
Post is signed
→ POST /api/posts/1/vote
✓ Vote recorded on post #1
→ POST /api/posts/2/vote
✓ Vote recorded on post #2
→ POST /api/posts/1/comments
✓ Comment #2 added to post #1
→ POST /api/posts/2/comments
✓ Comment #3 added to post #2
→ POST /api/messages/send
✓ Message sent from demo_bindhu to demo_alice
→ POST /api/messages/send
✓ Message sent from demo_alice to demo_bindhu
→ POST /api/messages
✓ Messages retrieved for demo_alice
→ POST /api/messages
✓ Messages retrieved for demo_bindhu
→ POST /api/feed
✓ Feed for demo_alice: 2 posts
Post is signed
Post is signed
→ POST /api/feed
✓ Feed for demo_bindhu: 2 posts
Post is signed
Post is signed
→ POST /api/subreddits/r/demo/leave
✓ demo_alice left r/demo
→ POST /api/subreddits/r/demo/leave
✓ demo_bindhu left r/demo
[Logged in as: bindhu]
Choice: d
RUNNING AUTOMATED DEMO
Step 1: Registering users...
✓ Registered demo_alice
✓ Registered demo_bindhu
Step 2: Logging in...
✓ Logged in demo_alice
✓ Logged in demo_bindhu
Step 3: Joining subreddits...
✓ demo_alice joined r/demo
✓ demo_bindhu joined r/demo
Step 4: Creating posts...
✓ demo_alice posted in r/demo
✓ demo_bindhu posted in r/demo
Step 5: Getting posts by ID...
✓ Retrieved post #1 successfully
✓ Retrieved post #2 successfully
Step 6: Voting on posts...
✓ demo_bindhu upvoted post #1
✓ demo_alice upvoted post #2
Step 7: Commenting on posts...
✓ demo_bindhu commented on post #1
✓ demo_alice commented on post #2
Step 8: Sending direct messages...
✓ demo_bindhu sent message to demo_alice
✓ demo_alice sent message to demo_bindhu
Step 9: Getting direct messages...


Terminal
Local
✓ POST /api/subreddits/:name/join - Join subreddit
✓ POST /api/subreddits/:name/leave - Leave subreddit
✓ POST /api/posts - Create new post
✓ GET /api/posts/:id - Get post by ID
✓ POST /api/posts/:id/vote - Vote on post
✓ POST /api/posts/:id/comments - Add comment
✓ POST /api/messages/send - Send direct message
✓ POST /api/messages - Get user messages
✓ POST /api/feed - Get user feed
[Logged in as: bindhu]
Choice: h
BONUS FEATURE: All posts can be cryptographically signed!
Available Commands:
1 - Register user (generates RSA-2048 keys)
2 - Login user
3 - Join subreddit
4 - Leave subreddit
5 - Create post (SIGNED if you have keys)
6 - View post (verifies signature) ✓
7 - Comment on post
8 - Vote on post (upvote/downvote)
9 - Get feed
10 - Send direct message
11 - View direct messages
d - Run automated demo
h - Show this help menu
q - Quit
[Logged in as: bindhu]
Choice: q
Goodbye! Thanks for using Reddit Clone.
bindhu@Bindhus-MacBook-Air reddit_gleam %

Run Multiple Clients:





Terminal Local

```
→ POST /api/register
  ℹ Generating RSA-2048 keypair for charlie...
→ POST /api/login
  ✓ Logged in: bindhu
  ✓ Registered: charlie with RSA-2048 public key
→ POST /api/subreddits/r/gleam/join
  ✓ alice joined r/gleam
→ POST /api/register
  ℹ Generating RSA-2048 keypair for diana...
→ POST /api/subreddits/r/gleam/join
  ✓ bindhu joined r/gleam
→ POST /api/login
  ✓ Logged in: charlie
  ✓ Registered: diana with RSA-2048 public key
→ POST /api/posts
 ✓ Post signed successfully
   ✓ Post #1 created by alice in r/gleam
→ POST /api/register
  ℹ Generating RSA-2048 keypair for eve...
  ✓ Registered: eve with RSA-2048 public key
→ POST /api/posts
 ✓ Post signed successfully
   ✓ Post #2 created by bindhu in r/gleam
→ POST /api/subreddits/r/gleam/join
  ✓ charlie joined r/gleam
→ POST /api/login
  ✓ Logged in: diana
→ GET /api/posts/1
  ✓ Post #1 retrieved
Post is signed
→ GET /api/posts/1
→ POST /api/login
→ POST /api/posts
  ✓ Post #1 retrieved
  ✓ Logged in: eve
 ✓ Post signed successfully
 Post is signed
  ✓ Post #3 created by charlie in r/gleam
→ POST /api/subreddits/r/gleam/join
  ✓ diana joined r/gleam
```

```
CLIENT: charlie

STEP 1: USER REGISTRATION
Client: charlie
Time: 21:06:38
Endpoint: POST /api/register
Response: {"status":"success","username":"charlie","public_key":"g2wAAAACbQAAAAMBAAFtAAABAJS7TFIcHzb86A/ppxDWaoU9eWcZViOz831N/m67DfZgSeSu3POsfhaguFjcBBGQCzryHhPkbBZbfuQrz492dwGu1a+vZTFU82fkfG4+AJt/XJc4VszteAf24UOC7YENEoHqRQHee97orfbGZQdQPZnISVUgd92kgcDYLCEv1Vp9svoQqpO4liixp+KY2Bv1T8lQWDVtZQBH5FclNZYIXDcN/e5K+hoBZokorRxZw8KK8WrLCjtxu5y0Fuvo2xheil8uXUEvdRydlAd/rPLyByT+6af7kbrRUgGHeKMEonQpMcbaZJUYUXbU4g6h4hWADOxL4ggRqqRzeOp27LKHvj9tAAABACNDLNsi07vDxnlPWk7Bp0ujwY7SEP5ynL4Oqc4zByJsvdg8JSKYuORAgrR+pXBMiHOXPk5VD9NVKgaKSJEM8zYw73TRztodXAA7ycyRYq97SMaJkvrPKnAo9kt9VTYNlV+73VSQHSxCWkxxyS4stq4p0Ekgg4rHIARKvFu5 7Ymgexxs8Yrwp9DIrhWWhnddyWAO8kDioS7rND9MiH9NAsEvATwABDMke2FjzF5g52xA4kZl+cY+l8xcHUQjjuF608F3rzbssNx uzG0ekIuLCEYcIwndU+lgUvFIh14B+57L5Hm2odeeUGOwnYvyhIwX7w8AiXeu4TxU8ZhWXGh/zhFtAAAAgMuWHsin8KPyCmem2m s5BRaixm+ioU8elgKBoTm8q6y9dhbX+l3s1pcnTk3gxTlq24jj8oDnc7QUIfj0E9NkKl5ntnqQ9SYwyrIpYSUgttJVRtCMK0T1Q OgnUNna0GRDOCfvqIkTxdinDisPWX+eLFc7uP4ikFXUC3de/cM+jDrFbQAAAIC7BdWkw9cwst9JfQRyRBusjJlsbiQOdnTFWw4W cYYjMoVghS1I7ALMYsGYsnrySHOn0qtcHTRm/Ka6nklBFi76l+F64JPj8vAKN6tFmoyt3XH4hwPG6Elx/eKV+IZ3aX0Ox5KpXe kHBu1ki3qnI3iDl1Q9lZyQLOyX9A1Gox1M20AACAgDyaPWhTkuF6Ho4iLMHo654ygmUdT0+FDCUdKfz8Sz0cIj1/DiS98s4ma7 W2ZXliuXJj8UBZSEgaMheEWXZRnc+USNYAJUM8mkaoU1us7QYcKgbwxmnSrYH+UoubElypBbbXISkcSWDI8uSjSliWs4gB+cpJw lZjtphWHeLLL8VtAAAAgKtXexZKvfnFxr/DI7GOKE57OYyXk70UVYMC6vQA6c2RWZTIahcG8sFhsNA+pI3G5Ywxb+9aWgw3fZrk OjzpsIkqbl1SGJffu+6/D9Mv9/ejEOshgmds/sRoYaeDQIhGAuawyt3EnsnhXpmISTxOVcbIHSIJrcVa9cbz9QLR4eD1bQAAAIC BvPZ3qHlsje3WLQY64HJiiNbnplpjz5tDl+MMnSiCe8Q/ZJEVSekiaMqoMYXsxjspqudR4mbCsiU2TvvPM3VbA75xqfg+seSacf u31EAHBQIebiYUsLkW6fJ7bltyUpqPKA5qgCJganXmnGUDm+hRzVNIC22/+Y1HBnX+/Fbpfmo=","warning":"SAVE YOUR PR IVATE KEY! You need it to sign posts. It won't be shown again."}
STEP 2: USER LOGIN
Client: charlie
Time: 21:06:39
Endpoint: POST /api/login
Response: {"status":"success","token":"BKti357U0fQ+C5oxvytb8BxKMbzLMPdoiAE+4t8xTW0=","username":"charlie"}
STEP 3: JOIN SUBREDDIT
Client: charlie
Time: 21:06:39
Endpoint: POST /api/subreddits/r/gleam/join
Response: {"status":"success","message":"Joined r/gleam"}
```

Terminal Local

```
  ✓ Vote recorded on post #1
→ POST /api/subreddits/r/gleam/join
  ✓ eve joined r/gleam
→ POST /api/posts/1/vote
  ✓ Vote recorded on post #1
→ POST /api/posts
→ GET /api/posts/1
 ✓ Post signed successfully
   ✓ Post #1 retrieved
   ✓ Post #4 created by diana in r/gleam
Post is signed
→ POST /api/posts/1/comments
  ✓ Comment #1 added to post #1
→ POST /api/posts/1/comments
→ POST /api/posts
  ✓ Comment #2 added to post #1
 ✓ Post signed successfully
   ✓ Post #5 created by eve in r/gleam
→ GET /api/posts/1
→ POST /api/posts/1/vote
  ✓ Post #1 retrieved
  ✓ Vote recorded on post #1
Post is signed
→ POST /api/messages/send
  ✓ Message sent from alice to bindhu
→ POST /api/posts/1/vote
→ GET /api/posts/1
→ POST /api/messages/send
→ POST /api/posts/1/comments
  ✓ Vote recorded on post #1
  ✓ Post #1 retrieved
  ✓ Message sent from bindhu to alice
  ✓ Comment #3 added to post #1
Post is signed
→ POST /api/messages
  ✓ Messages retrieved for alice
→ POST /api/messages/send
  ✓ Message sent from charlie to alice
→ POST /api/messages
→ POST /api/posts/1/vote
```

```
CLIENT: diana

STEP 1: USER REGISTRATION
Client: diana
Time: 21:06:38
Endpoint: POST /api/register
Response: {"status":"success","username":"diana","public_key":"g2wAAAACbQAAAAMBAAFtAAABAMGdTGEDjGuc3G1XaVZXj7nKqGbvkSEGAeU8TaMaduPcY84n/QPJ/rTuYpM7r6p3BFtvh+InC/E6bI5uGTlGHQK5/JF66MLIWriyVfu2VrHV4vMJQEYxfjv6tbWUYFUWISVtRuwFBy4xh7oWNrAUY1XB+/W2KV5LZC9+Ukt0c4hVx8SNjuPEYErDn3Z/D60Bmc/i2LIs673ZL7wzRNQnOM4akRaJIwwbMMdKUwdzWlcoLxRJKUOVxLfwW6mapgjvebGUDPloDlbqgSosy3Jp9k8PkYDCWT4pYtpj6VIswJ4g73F3XLtRAG2DV+ShbWvpvuh1yIkdRtZpHX66S/622+Vq","private_key":"g2wAAAAIbQAAAAMBAAFtAAABAMGdTGEDjGuc3G1XaVZXj7nKqGbvkSEGAeU8TaMaduPcY84n/QPJ/rTuYpM7r6p3BFtvh+InC/E6bISuGTlGHQK5/JF66MLIWriyVfu2VrHV4vMJQEYxfjv6tbWUYFUWISVtRuwFBy4xh7oWNrAUY1XB+/W2KV5LZC9+UktOc4hVx8SNjuPEYErDn3Z/D60Bmc/i2LIs673ZL7wzRNQnOM4akRaJIwwbMMdKUwdzWlcoLxRJKUOVxLfwW6mapgjvebGUDPloDlbqgSosy3Jp9k8PkYDCWT4pYtpj6VIswJ4g73F3XLtRAG2DV+ShbWvpvuh1yIkdRtZpHX66S/622+VtAAABAAPWH+JR1XsSAG6Mnqi4UwyMAhEJsQCytsaAQf8rK60eLwtLUMC93hv45HklMJ9NfNRqMSR/VSML7lGJx8D4k7CVinZATwZoPYWO+xP7klvM8JJ/EQf5b3y1KtthoIAR7TvG9SXOGUzs0rl5B5v7lQno+xrstIS+AUrNDcif0e6jA6yoKSOMPDT7OaNAHnUFSCVATYF44NfqgDY2u7UhVhlxHqljdZdsJ29LeaIYhcELujvF1YdhjXoiLICMZhb8gSBeamYnXd1OjqyTYNO4r9uKEvo7hiDZ3urC9beAqoaOPYtCv4jOP8ahdjWdjz7+a3SVRHX4n1Qnglv5CEZePy1tAAAAgN8P5H1UDL1Cu4ZRYiHTa4R+efMCE/13zz2BDHU5UhC7sg3H7wwuGQ6OHTQ4zkUeBmuM359zAm4HZhw7/Ue0Y8ebqdyLw0pQqpaoms8r5oEtNQllBCeIRZomGRiaSSTQLkB+DMHp3nzFYpNcdS96pIoz0YHl6ZYotfdWDiIJF52rbQAAAIDeNDzVsBWt6PTu0ycRQBlNohFqwlryxPcxcxHuG95qgsWKV9fqODpsnNOETEzcS/ZCb070TeXtSLfvzQIi9Ly2mnkAGMON/qzlbukylhgkJNFg4u2h9t2jwTWMdJKNzbfWrGPdGqqpfbJxXW4yxEVksBVYGJluhkPnI2QU8do8r20AAACAH5CmKr4MF3QsCTPXCSZb3qHvK94taf7cSr3KP4RfApMs/Ro/zS9uWErwAwXQjVtUWT/UHUIvI9bUMka17tSuEf8LKHR2rDuCCFPPpGV39vupAaAa2GVGRCwD/SFeL7lt8h5kO8SWgV2FN4ksPFskkZIHobQhO+i C5387/MEAoIvtAAAAgH9FCkcWiQRINuoHtQuzXArUNgeg9Hiyiicfkni AK99zsXNwpvG3bPdCx/+xuCBBvfYFTQKvD2/eQ8Qs9v lQljo8tetxQ6iz4xGyeg0IXSFumSJWN+eZvw8JHgQpzTs9RfvGV5iPqopTvX4jFngrZNer5z3DMTZV3tSnt64rbGVvbQAAAICWQ b6NzKpLPvZBPfzY9J/FbCogYBE1idfCgJQ9XDVx/Pqk87+RMD9tUKuuwyYrt0zHL3cssGnD+9PB8krsVhBFixbh/IMCPT0fWEd5 X+dKza75+X0EPmFfNOmZEOFiEs1FkVcGcSq25HGwa1PTwFEh93ilpVqdJZ8t3OqpBI0mCmo=","warning":"SAVE YOUR PRIV ATE KEY! You need it to sign posts. It won't be shown again."}
STEP 2: USER LOGIN
Client: diana
Time: 21:06:39
Endpoint: POST /api/login
Response: {"status":"success","token":"5d0cKx4JPHxJcolbiuAqbUm8WjWWdvrTKMi9OjJlHqs=","username":"diana"}
STEP 3: JOIN SUBREDDIT
Client: diana
Time: 21:06:40
Endpoint: POST /api/subreddits/r/gleam/join
Response: {"status":"success","message":"Joined r/gleam"}
```

8. Conclusion

This project represents a complete end-to-end implementation of a Reddit-like social platform, built entirely from scratch with a strong focus on system design, modular architecture, and real-world functionality. The engine, REST API, client interface, and simulator work together to deliver a cohesive ecosystem where users can register, join communities, create posts, comment, vote, exchange messages, and retrieve personalized feeds.

A significant highlight of the project is the integration of a full **RSA-2048 digital signature framework**, which adds a layer of authenticity and security to user-generated content. By enabling posts to be signed and verified, the system mirrors the trust mechanisms used in secure communication and demonstrates the practical application of cryptographic principles.

The project's architecture separates concerns cleanly: the engine handles business logic, the REST API exposes a modern interface, and the client/simulator ensures usability and scalability testing. This separation not only improves clarity but also makes the system easier to extend in the future.

Overall, the project successfully fulfills all mandatory requirements and excels beyond them by delivering the optional cryptographic bonus feature. It showcases an understanding of distributed systems, concurrency, security, API design, and client–server communication. The work provides a strong foundation for future improvements - such as persistent storage, enhanced authentication, or UI development - and stands as a robust demonstration of both system-building and software engineering skills.