Pork-Professional-Networking

Repo:https://github.com/Jayashree-jannu/Prok-Professional-Networking.git

INTERNSHIP PROGRESS REPORT

Name: Jayashree S
Project Name: *Prok - Professional Networking*
Duration: 2 Months (Ongoing – 2 Weeks Completed)
Mentor: Mr. Aatham Ansari
Type: Team Project
Platform: Ubuntu Linux using Cursor IDE

Project Overview

The project *Prok - Professional Networking* is a full-stack web application development internship involving 16 modules. As of now, I have successfully completed 6 out of 7 assigned modules. The 7th module is currently in progress.

This internship mainly focuses on creating a scalable, modular professional networking platform, and it has given me hands-on experience in both frontend and backend development.

Key Tools & Technologies Used

- **OS**: Ubuntu (Linux)
- **IDE**: Cursor
- Version Control: Git, GitHub
- Frontend: HTML, CSS, JS (with npm modules)
- Backend: Python (with virtual environment and dependencies)
- **Terminal Commands**: Git, Node, npm, pip

What I Learned

- Using Linux terminal for efficient system control and project execution.
- Git commands for version control:
  - `git pull origin master`
  - `git checkout -b module-name`
  - `git push origin branch-name`

Installing frontend dependencies:

bash
CopyEdit
```
cd app/frontend
npm install
```

-

Installing backend dependencies and activating Python virtual environment:

bash

```
cd ../backend
python -m venv venv
source venv/bin/activate
pip install -r requirements.txt
```

-
- Running backend and frontend in separate terminals to view live output.
- Completed modules like:
  - Login/Signup UI
  - Profile Editing

    - Post Upload

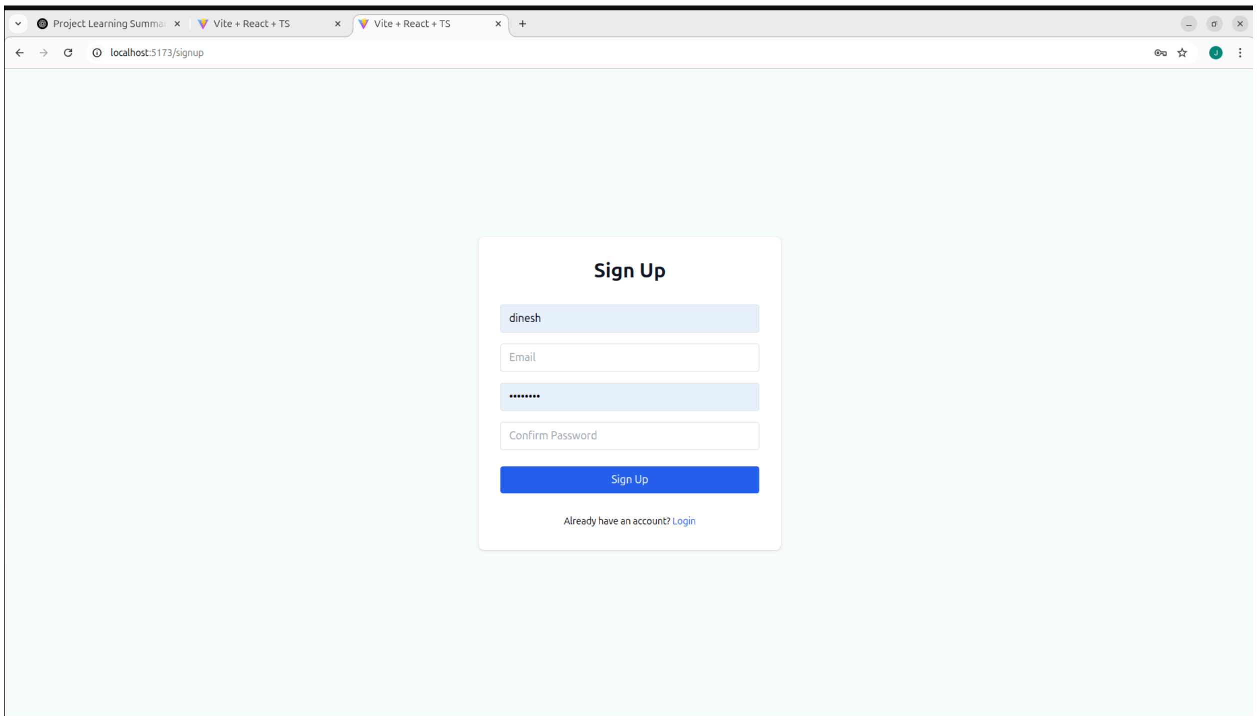
- Installing necessary libraries and dependencies for each module.
- Understood the importance of step-by-step task execution in development.

Conclusion

This internship has helped me build strong foundational skills in full-stack development, version control, and using Linux systems efficiently. I'm confident that the remaining modules will help me grow further and complete the project successfully under the guidance of my mentor.

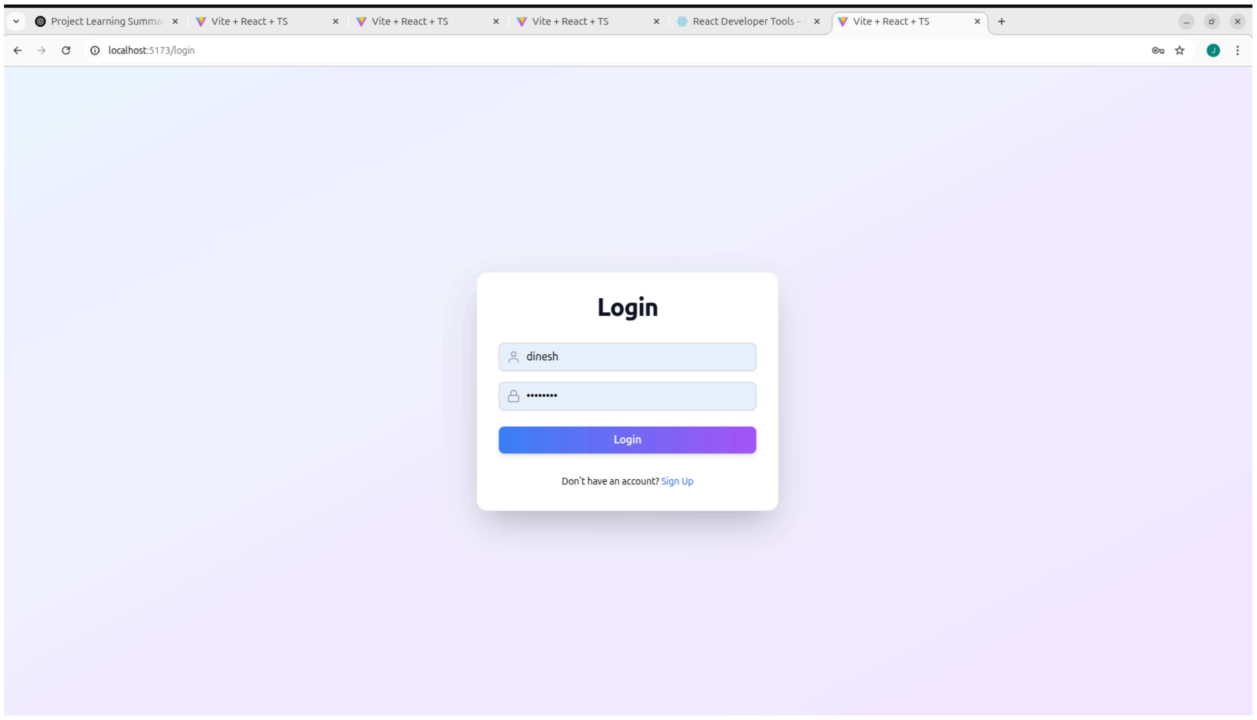
MODULE 1:

Signup page:



Signup Page: Complete registration system with user-friendly form, password complexity validation, real-time feedback, and seamless integration with backend authentication API. Both pages are fully responsive and connected to the Flask backend running on port 5000.

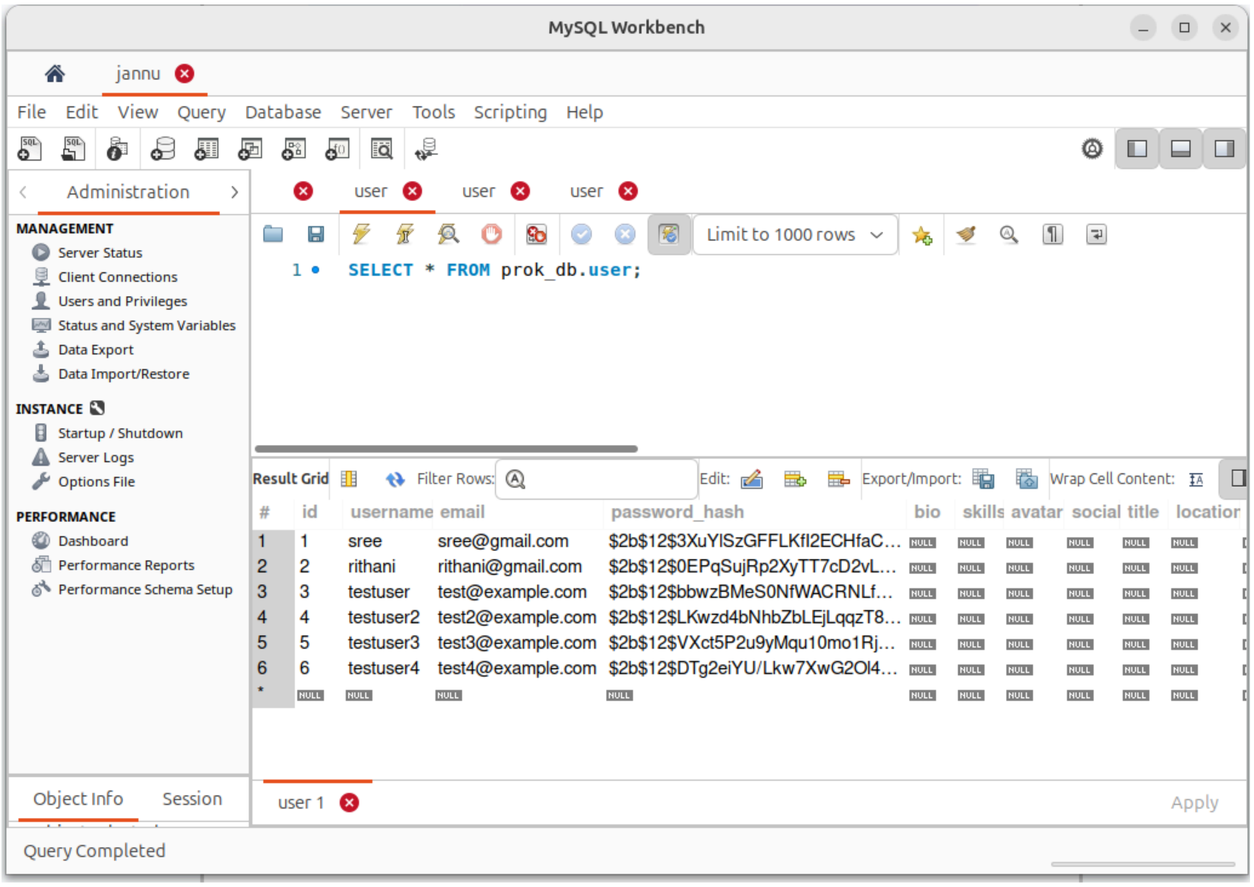
Login Page:



Login Page: Successfully implemented with clean, professional design featuring username/email input, password field, form validation, error handling, and navigation to signup. Removed all testing shortcuts (logout, auto-login, direct login) for production-ready functionality.

Module 2:

Authentication Backend:



```
SELECT * FROM prok_db.user;
```

#	id	username	email	password_hash	bio	skills	avatar	social	title	locatior
1	1	sree	sree@gmail.com	$2b$12$3XuYlSzGFFLKfI2ECHfaC…	NULL	NULL	NULL	NULL	NULL	NULL
2	2	rithani	rithani@gmail.com	$2b$12$0EPqSujRp2XyTT7cD2vL…	NULL	NULL	NULL	NULL	NULL	NULL
3	3	testuser	test@example.com	$2b$12$bbwzBMeS0NfWACRNLf…	NULL	NULL	NULL	NULL	NULL	NULL
4	4	testuser2	test2@example.com	$2b$12$LKwzd4bNhbZbLEjLqqzT8…	NULL	NULL	NULL	NULL	NULL	NULL
5	5	testuser3	test3@example.com	$2b$12$VXct5P2u9yMqu10mo1Rj…	NULL	NULL	NULL	NULL	NULL	NULL
6	6	testuser4	test4@example.com	$2b$12$DTg2eiYU/Lkw7XwG2Ol4…	NULL	NULL	NULL	NULL	NULL	NULL
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Query Completed

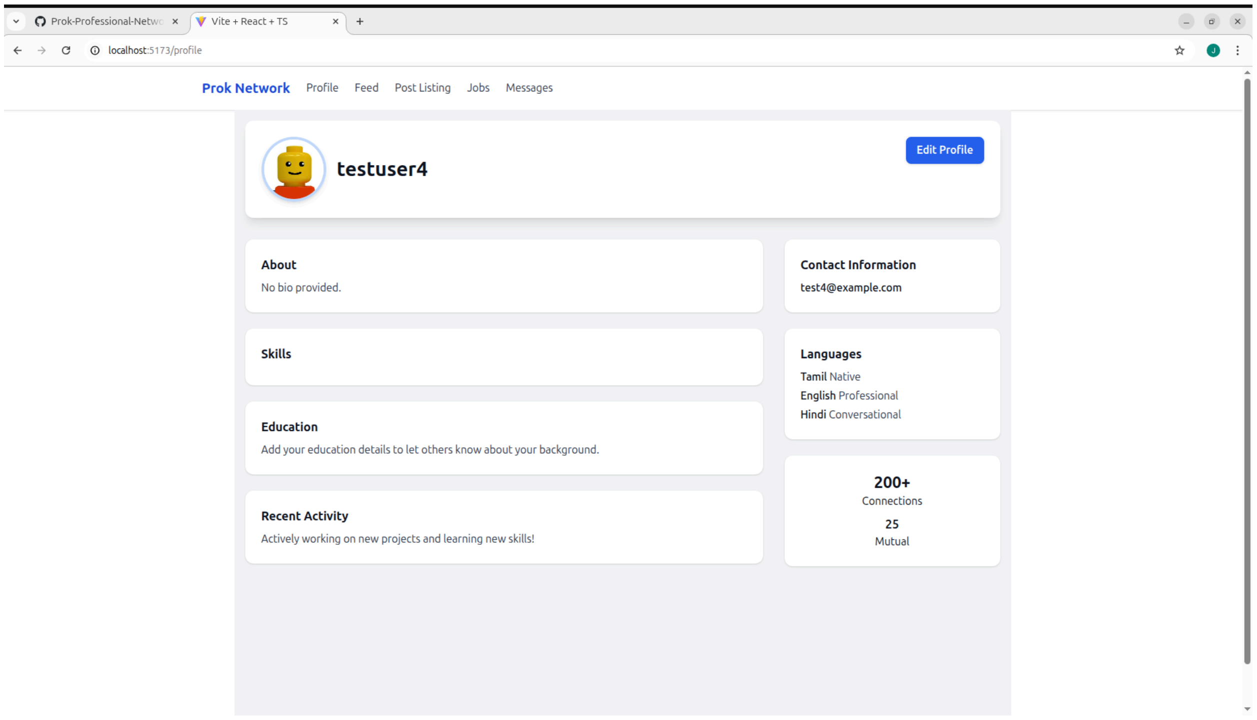
Database Setup: Created User model with all profile fields (username, email, bio, skills, etc.) and set up SQLAlchemy database with proper migrations.

Login/Signup API: Built secure authentication endpoints with password hashing, JWT token generation, and input validation for user registration and login.
Profile Management: Implemented RESTful API for profile CRUD operations including image upload functionality and real-time profile updates.

Security Features: Added CORS configuration, rate limiting, password complexity validation, and proper JWT token management with configurable expiration.

Error Handling: Implemented comprehensive error handling for duplicate users, invalid credentials, and proper HTTP status codes for all authentication operations.

Module 3:
Profile View & Edit UI:



testuser4

About
No bio provided.

Contact Information
test4@example.com

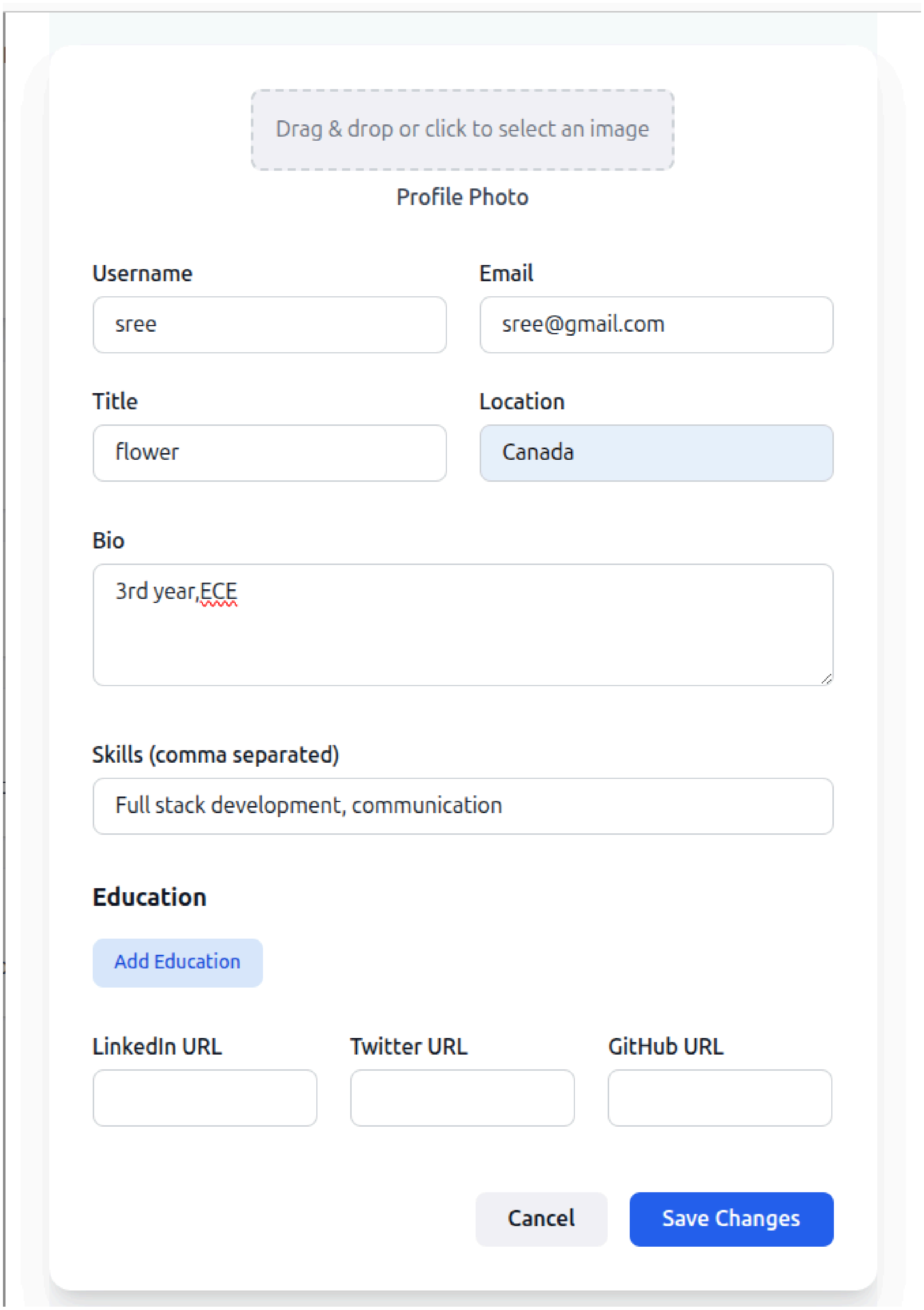
Skills

Languages
Tamil Native
English Professional
Hindi Conversational

Education
Add your education details to let others know about your background.

200+
Connections
25
Mutual

Recent Activity
Actively working on new projects and learning new skills!



Profile Photo

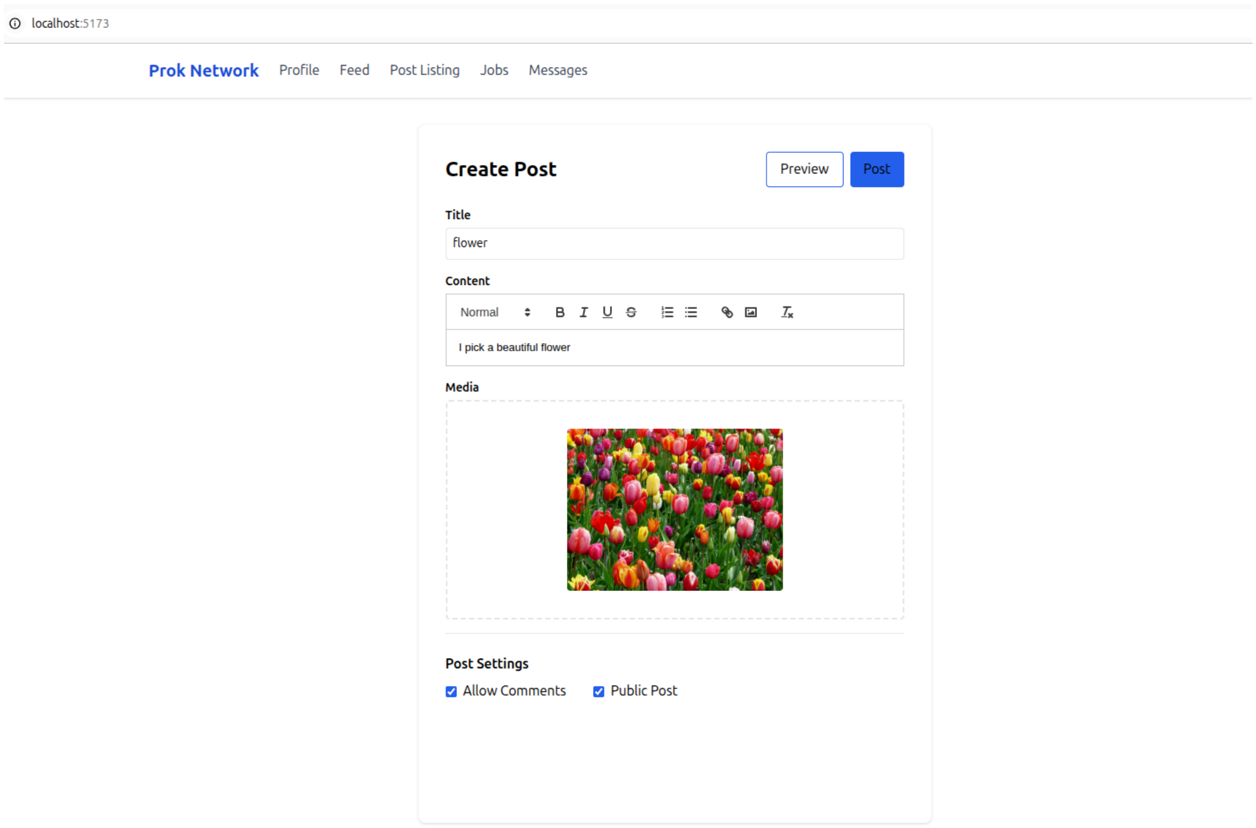
Username	sree
Email	sree@gmail.com
Title	flower
Location	Canada
Bio	3rd year,ECE
Skills (comma separated)	Full stack development, communication

Education

LinkedIn URL	
Twitter URL	
GitHub URL	

Module 5:

Post creation:



Create Post

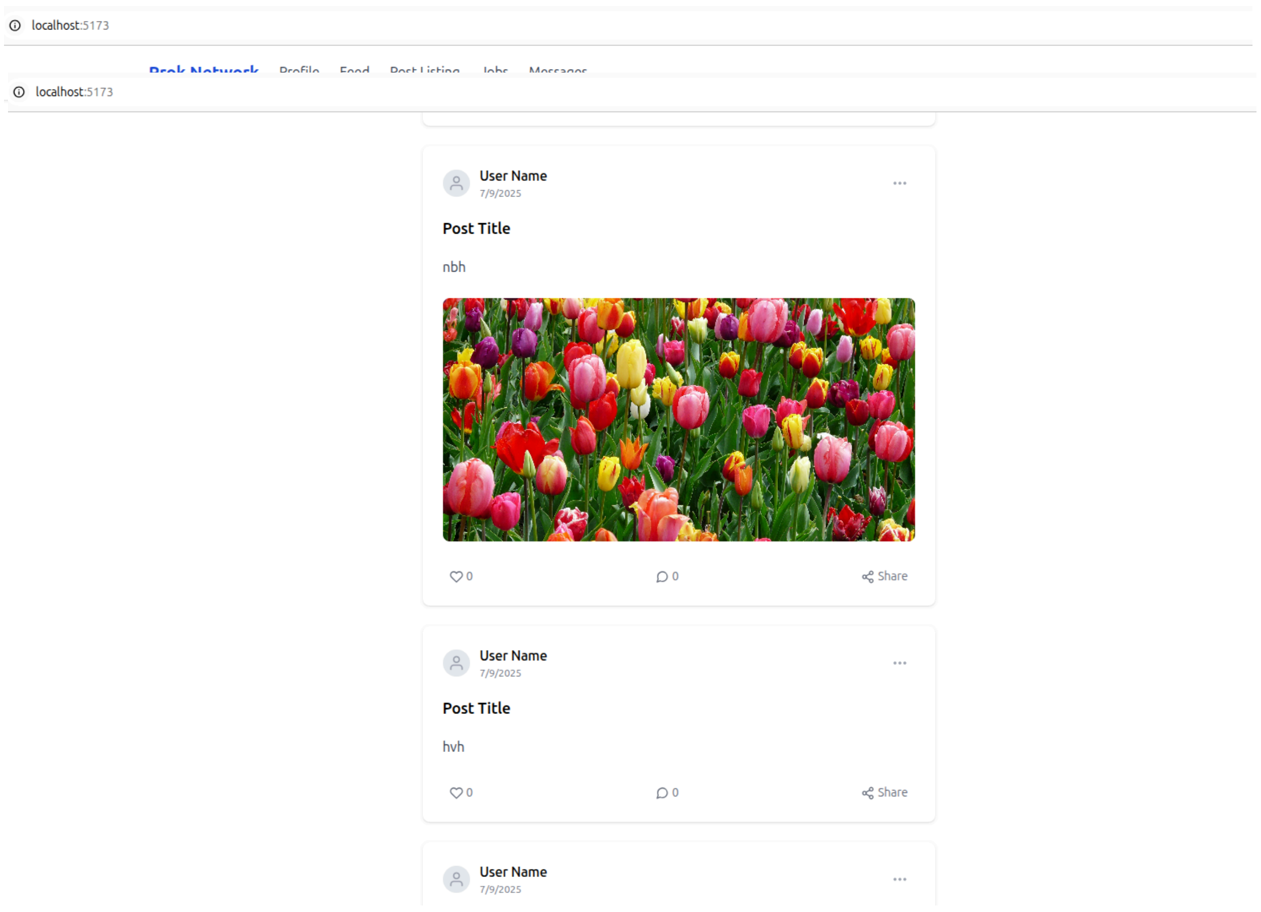
Title: flower

Content: I pick a beautiful flower

Media

Post Settings

[✓] Allow Comments [✓] Public Post



localhost:5173

Prok Network Profile Feed Post Listing Jobs Messages

localhost:5173

User Name
7/9/2025

Post Title

nbh

0 0 Share

User Name
7/9/2025

Post Title

hvh

0 0 Share

User Name
7/9/2025

Module 1: Login/Signup UI

Error 1: npm ERR! Missing script: "dev"

Solution: cd app/frontend && npm run dev
Difficulty: Wrong directory - needed to be in frontend folder
Error 2: Frontend connection refused to backend
Solution: Updated API URLs from port 8000 to 5000 in frontend
Difficulty: Port mismatch between frontend config and backend

Module 2: Authentication Backend
Error 1: ModuleNotFoundError: No module named 'flask_cors'
Solution: source venv/bin/activate && pip install flask-cors
Difficulty: Virtual environment not activated, missing dependency
Error 2: Address already in use on port 5000
Solution: pkill -f python or use different port
Difficulty: Multiple backend instances running
Error 3: Database connection errors
Solution: Used local MySQL instead of hardcoded URI
Difficulty: Database configuration issues

Module 3: Profile View & Edit UI
Error 1: Frontend hot reload issues
Solution: rm -rf node_modules/.vite && npm run dev
Difficulty: Vite cache conflicts
Error 2: Profile image upload failures
Solution: Added cache-busting timestamps to image URLs
Difficulty: Browser caching preventing new images

Module 4: Profile Edit Backend
Error 1: JWT token validation failures
Solution: Proper token format in Authorization header
Difficulty: Authentication middleware issues
Error 2: File upload size limits
Solution: Increased max content length in Flask config
Difficulty: Large image uploads failing

Module 5: Post Creation
Error 1: CORS errors on file upload

Solution: Added proper CORS headers for multipart/form-data
Difficulty: Cross-origin file upload restrictions
Error 2: Media file storage issues
Solution: Created proper upload directories and permissions
Difficulty: File system access and storage paths
Module 6: Posts Listing
Error 1: Infinite scroll not working
Solution: Fixed pagination logic and scroll detection
Difficulty: React state management for pagination
Error 2: Image loading errors
Solution: Added error handling for broken image URLs
Difficulty: Network issues and missing media files
Common Commands Used:

```
# Backend
cd app/backend && source venv/bin/activate && python main.py

# Frontend
 cd app/frontend && npm run dev

 # Kill processes
 pkill -f python

  # Clear cache
  rm -rf node_modules/.vite
 Apply to Prok-Profess...
 Run
```

Key Difficulties Overcome:
Environment Management: Virtual environment activation and dependency installation
Port Conflicts: Managing multiple services on different ports
CORS Issues: Cross-origin requests between frontend and backend
File Uploads: Proper handling of multipart data and file storage
Authentication: JWT token management and validation
State Management: React component state and API integration
Cache Issues: Browser and build tool caching problems

Database Integration: Connection and query optimization