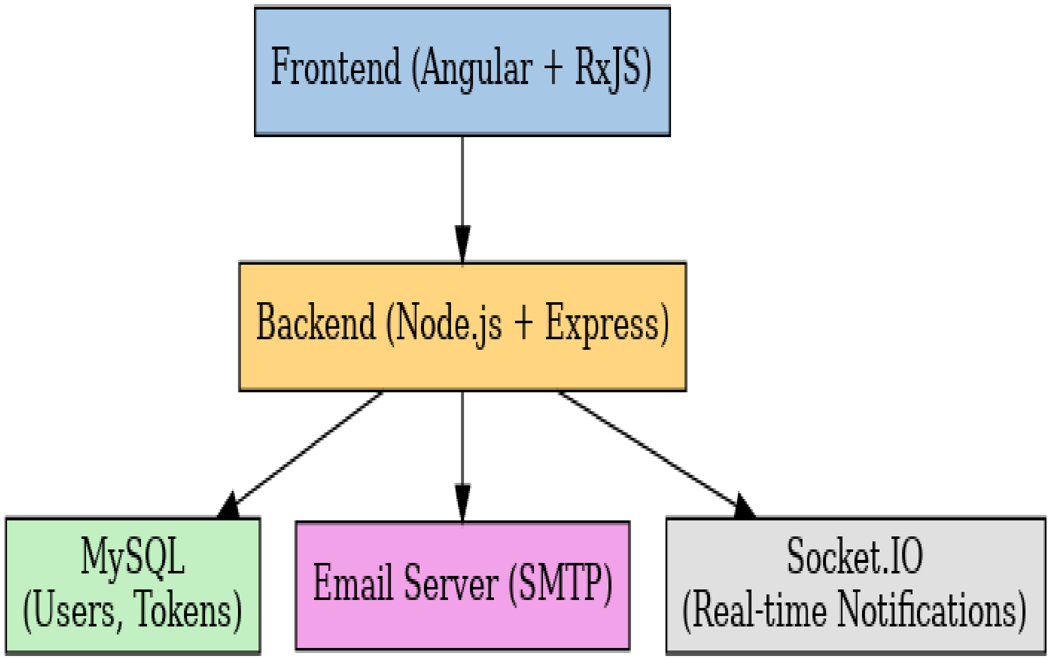
# Authentication & User Management App Documentation

## See diagram below for architecture:


Frontend (Angular + RxJS)
Backend (Node.js + Express)
MySQL (Users, Tokens)
Email Server (SMTP)
Socket.IO (Real-time Notifications)

# Authentication & User Management App Documentation

## Overview
This full-stack application provides a secure authentication and user management system with real-time updates and admin analytics.
It allows users to register, verify email, log in, reset passwords, and manage their accounts securely.
It provides administrators with a dashboard to view user activity, revoke sessions, and monitor usage statistics.

## User Features
- Account creation with email verification
- Secure login/logout
- Password reset via email
- 10-minute inactivity timeout
- Account deletion with analytics logging
- Cookie-based refresh token handling

## Admin Features
- Dashboard for all users with timestamps and activity
- Filter users by date range, name, or status

- Revoke refresh tokens
- Delete or deactivate users
- Real-time activity charts (Socket.IO)

## Security Architecture
- Password hashing (bcrypt)
- JWT authentication + cookie-only refresh tokens
- CSRF protection with SameSite cookies
- SQL injection prevention with parameterized queries
- HTTPS enforcement in production
- Role-based admin route protection

## Technologies Used

| Layer | Technology | Purpose |
|-------|-------------|----------|
| Frontend | Angular 15, RxJS, Socket.IO Client | UI & real-time updates |
| Backend | Node.js + Express, Socket.IO, Nodemailer | API, real-time, emails |
| Database | MySQL | Persistent user and token storage |
| Auth | JWT + bcrypt | Authentication |
| DevOps | GitHub Actions, Shell Deploy Script | CI/CD |
| Email | HTML templates | Email verification & reset |

## System Flow
1. User registers → verification email sent
2. User verifies email → account activated
3. User logs in → JWT + refresh token issued
4. Inactivity (10 min) → logout
5. Admin views users and activity in dashboard
6. Real-time data via Socket.IO

## Flowchart
See flowchart below for architecture overview.