**Team:** Santiago Macias Fernandez, Oscar Juanpablo Rivera León, Marco Rodrigo Campoy Mendoza

**TECHNOLOGY:**
- Runtime: Node.js
- Web Framework: Express.js
- Template Engine: EJS
- HTTP Client: Axios
- Authentication: Passport.js with Google OAuth 2.0
- Database: MongoDB with Mongoose ODM
- Session Management: Express-session

**ARCHITECTURE & LOGIC:**

**1. SERVER INITIALIZATION & CONFIGURATION:**
- Creates Express application with environment-aware settings
- Sets port from environment variable (default: 3001)
- Configures EJS as template engine with views directory
- Serves static files from "public" directory
- Implements trust proxy for production deployment on Render

**2. DATABASE ARCHITECTURE:**
**MongoDB Integration:**
  * userId (String, unique) - Google OAuth user ID
  * displayName (String) - User's display name
  * email (String) - User's email
  * clickedAt (Date) - Timestamp of user action

**- Click Counter:**
  * count (Number) - Global counter for "Join the Cause" feature

**Database Features:**
- Graceful fallback when MongoDB unavailable
- Automatic counter initialization
- Connection status monitoring

**3. AUTHENTICATION SYSTEM:**
**Google OAuth 2.0 Implementation:**
- Passport.js strategy for Google authentication
- Session-based user serialization/deserialization
- Secure cookie configuration with environment settings
- Authentication status API endpoints

**Routes:**
- GET /auth/google - Initiates OAuth flow
- GET /auth/google/callback - OAuth callback handler
- GET /logout - Session termination
- GET /api/auth/status - Authentication status check

**4. ROUTING STRUCTURE:**

**PRIMARY ROUTES:**
**- GET / (Main Page):**
  * Fetches random quotes from ZenQuotes API
  * Error handling with fallback quotes
  * Renders dynamic content using EJS template
  * Displays ocean conservation statistics

**- GET /help (Help & Action Page):**

  * Shows user participation statistics
  * Implements "Join the Cause" counter
  * Prevents duplicate user participation
  * Renders organization information

**API ENDPOINTS:**

- GET /api/user - Returns user authentication status
- POST /api/click - Registers user participation (requires auth)
- GET /api/click-count - Returns participation statistics
- GET /api/auth/status - Authentication verification
- GET /health - System health check

**5. DATA FLOW & INTEGRATION:**

**External API Integration:**

- ZenQuotes API for daily inspirational quotes
- Asynchronous HTTP requests with Axios

**Database Operations:**

- User participation tracking with duplicate prevention
- Atomic counter increments for real-time statistics
- Connection state management with graceful degradation

**6. ENVIRONMENT CONFIGURATION:**

**Multi-environment Support:**

- Development: Localhost with relaxed security
- Production: Render deployment with secure settings
- Dynamic configuration based on NODE_ENV
- Conditional database connection handling

**Environment Variables:**

- PORT - Server port configuration
- SESSION_SECRET - Session encryption key
- GOOGLE_CLIENT_ID - OAuth application ID
- GOOGLE_CLIENT_SECRET - OAuth application secret
- MONGODB_URI - Database connection string
- NODE_ENV - Environment specification

**7. ERROR HANDLING & RESILIENCE:**

Comprehensive Error Management:

- Database connection failures with fallbacks
- API integration errors with default content
- Authentication failures with proper redirects
- User input validation and duplicate prevention
- Console logging for development
- Structured error pages for clients

**Fallback Mechanisms:**

- Local storage for counter when database unavailable
- Default quotes when external API fails
- Continued functionality without authentication
- Session management with secure defaults

## 8. KEY FEATURES IMPLEMENTED:

**User Management:**
- Google OAuth 2.0 authentication
- Session-based user tracking
- Profile display with user information
- Secure logout functionality

**Engagement Features:**
- "Join the Cause" participation system
- Real-time counter updates
- Duplicate participation prevention
- User commitment tracking

**Content Delivery:**
- Dynamic quote generation
- Responsive template rendering
- Static asset serving
- API-driven content updates

## 9. DEPENDENCIES MANAGEMENT:

**Core Dependencies:**
- express - Web server framework
- ejs - Dynamic HTML templating
- axios - HTTP client for external APIs
- passport - Authentication middleware
- passport-google-oauth20 - Google OAuth strategy
- mongoose - MongoDB object modeling
- express-session - Session management

**Development Dependencies:**
- nodemon - Development auto-restart

## 10. SECURITY IMPLEMENTATIONS:

**Session Security:**
- Environment-specific cookie settings
- HTTP-only cookies
- Same-site policy enforcement
- Session expiration management

**Authentication Security:**
- OAuth 2.0 protocol compliance
- Secure token handling
- User session validation
- Protected API endpoints

## 11. SERVER LIFECYCLE:

**1. Initialization Phase:**

- Load environment configuration
- Initialize Express application
- Configure middleware and view engine
- Establish database connection

**2. Runtime Phase:**

- Handle incoming HTTP requests
- Manage user sessions and authentication
- Process API calls and database operations
- Serve dynamic and static content

**3. Maintenance Phase:**

- Health monitoring and status reporting
- Error handling

## 12. DEPLOYMENT:

**Production Features:**

- Environment-aware configuration
- Health check endpoints
- Proxy trust configuration
- Secure session management
- Database connection pooling
- Comprehensive error handling

**Development Features:**

- Hot reload with nodemon
- Detailed console logging
- Local database support
- Easy configuration setup