**Software Requirements Specification (SRS) Document**

**Project Title**: Full-Stack Video Streaming Application (YouTube Clone)  
**Version**: 1.0  
**Prepared by**: [Your Name]  
**Date**: [DD/MM/YYYY]

**1. Introduction**

**1.1 Purpose**

This document outlines the functional and non-functional requirements for a YouTube-like video streaming platform built with **Next.js, Node.js, MongoDB, and Bunny CDN**.

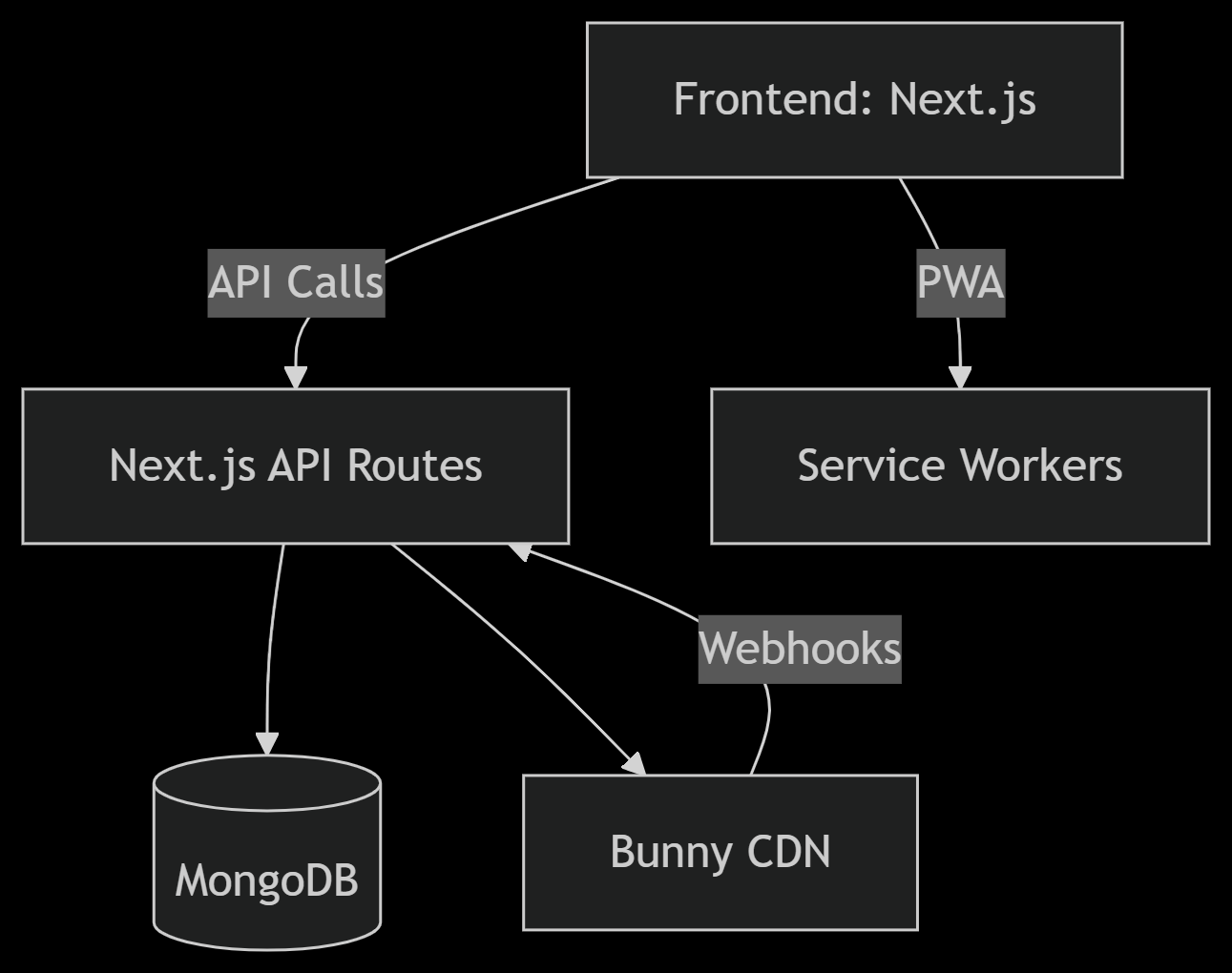
**1.2 Scope**

The application will include:

* User authentication (sign-up/login).
* Video upload/processing (Bunny CDN).
* Multi-resolution playback (240p to 1080p).
* User analytics (views, watch time, demographics).
* Real-time notifications (comments, subscriptions).
* SEO optimization and PWA support.

**2. Overall Description**

**2.1 System Architecture**

**2.2 User Roles**

| **Role** | **Permissions** |
| --- | --- |
| Guest | Browse videos, search, preview (no upload/comment) |
| User | Upload videos, comment, like, subscribe, view analytics |
| Admin | Manage categories, moderate content |

**3. Functional Requirements**

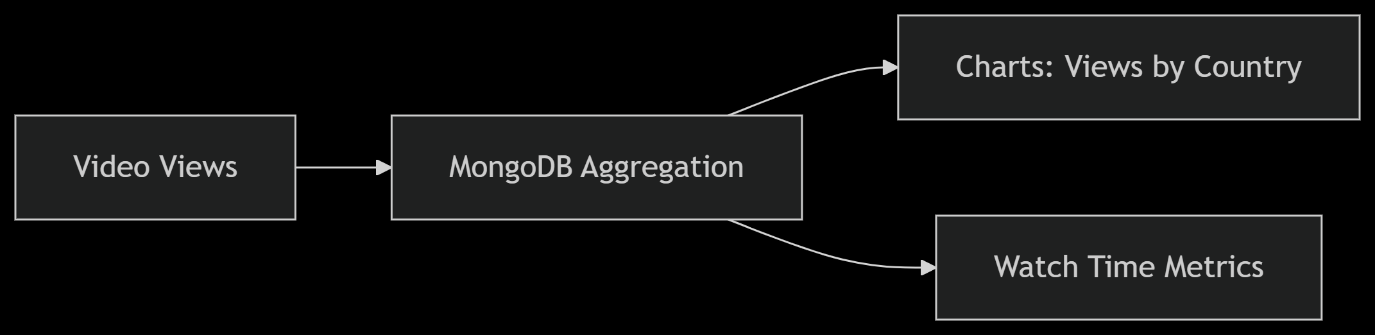
**3.1 User Authentication**

| **ID** | **Requirement** |
| --- | --- |
| FR1 | Users can sign up with email/password. |
| FR2 | Users can log in and persist sessions (JWT). |
| FR3 | Password hashing (bcrypt). |

**3.2 Video Management**

| **ID** | **Requirement** |
| --- | --- |
| FR4 | Upload videos to Bunny CDN (multi-resolution encoding). |
| FR5 | Save video metadata (title, description, tags) in MongoDB. |
| FR6 | Track watch history and resume playback. |

**3.3 Analytics Dashboard**



| **ID** | **Requirement** |
| --- | --- |
| FR7 | Display views, likes, shares per video. |
| FR8 | Show watch time trends (daily/weekly). |

**3.4 Real-Time Features**

| **ID** | **Requirement** |
| --- | --- |
| FR9 | WebSocket notifications for comments/subscriptions. |
| FR10 | Optimistic UI updates for likes/comments. |

**4. Non-Functional Requirements**

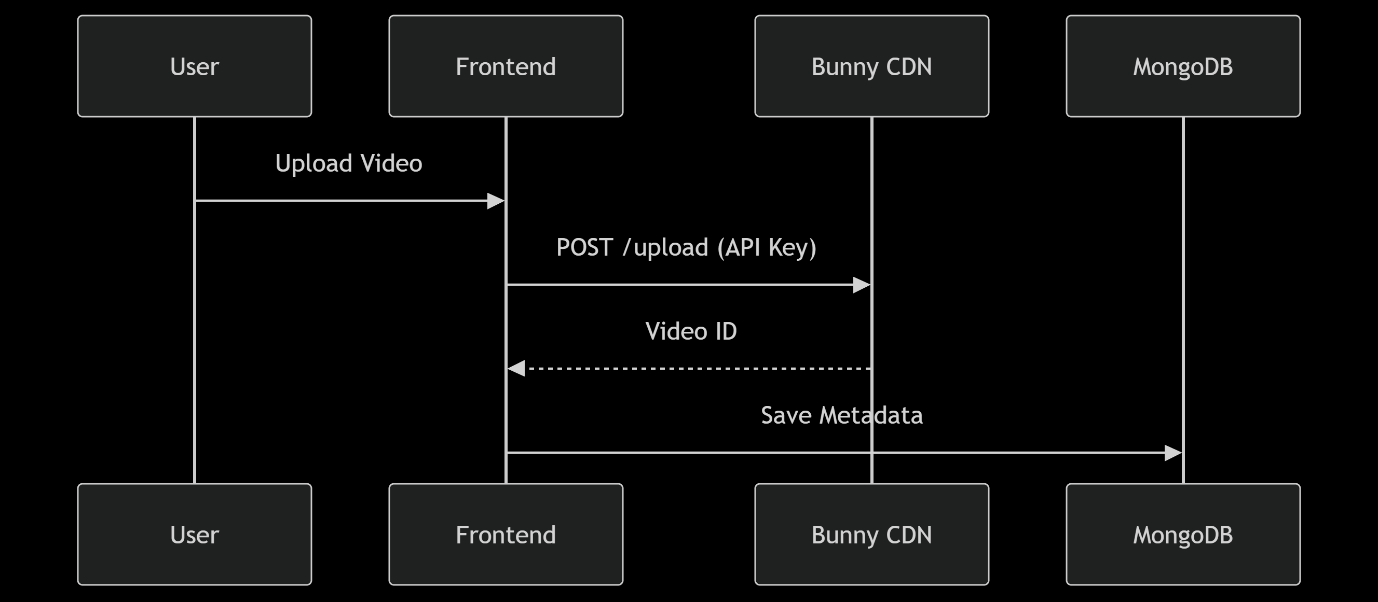
| **ID** | **Requirement** |
| --- | --- |
| NFR1 | Responsive design (mobile/desktop). |
| NFR2 | SEO-friendly (OpenGraph meta tags). |
| NFR3 | PWA support (offline caching). |
| NFR4 | Video load time <2s (Bunny CDN optimization). |

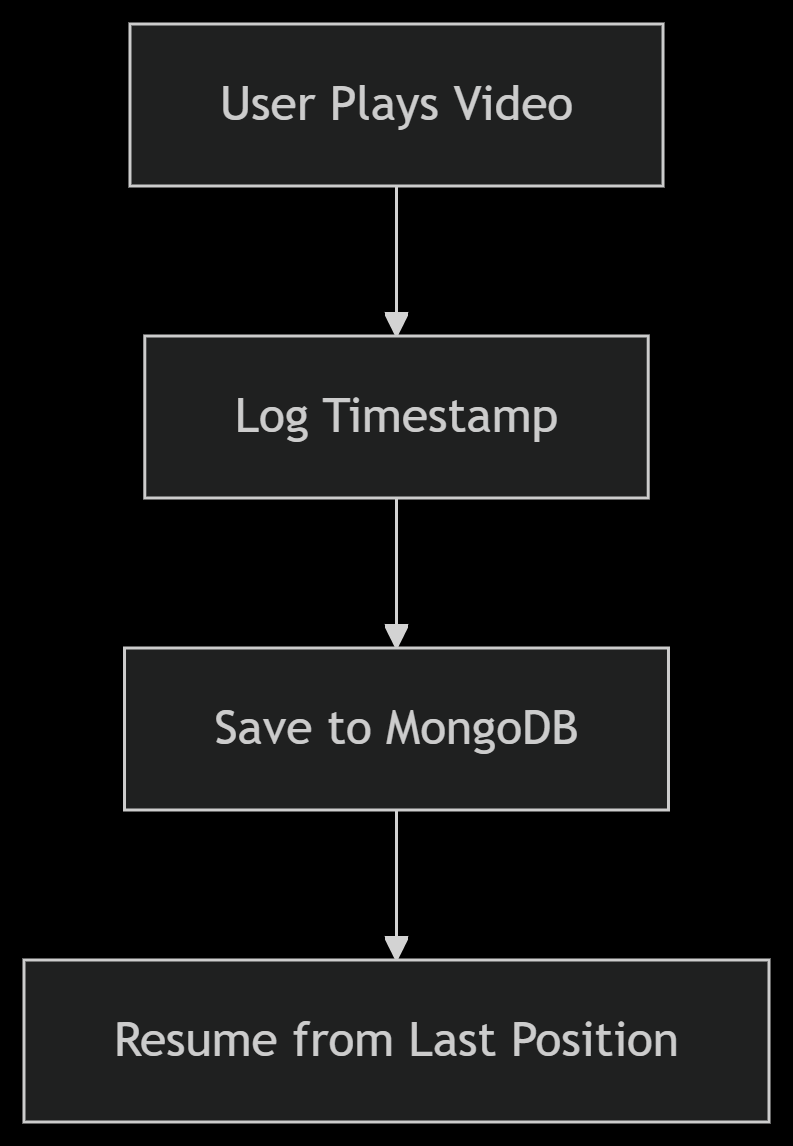
**5. Technical Stack**

| **Component** | **Technology** |
| --- | --- |
| Frontend | Next.js, Material-UI |
| Backend | Next.js API Routes, Node.js |
| Database | MongoDB (Atlas) |
| Video Processing | Bunny CDN (Stream + Storage) |
| Auth | NextAuth.js |

**6. Data Flow Diagrams**

**6.1 Video Upload Flow**

**6.2 Watch History Tracking**



**7. Use Case Diagrams**

Diagram

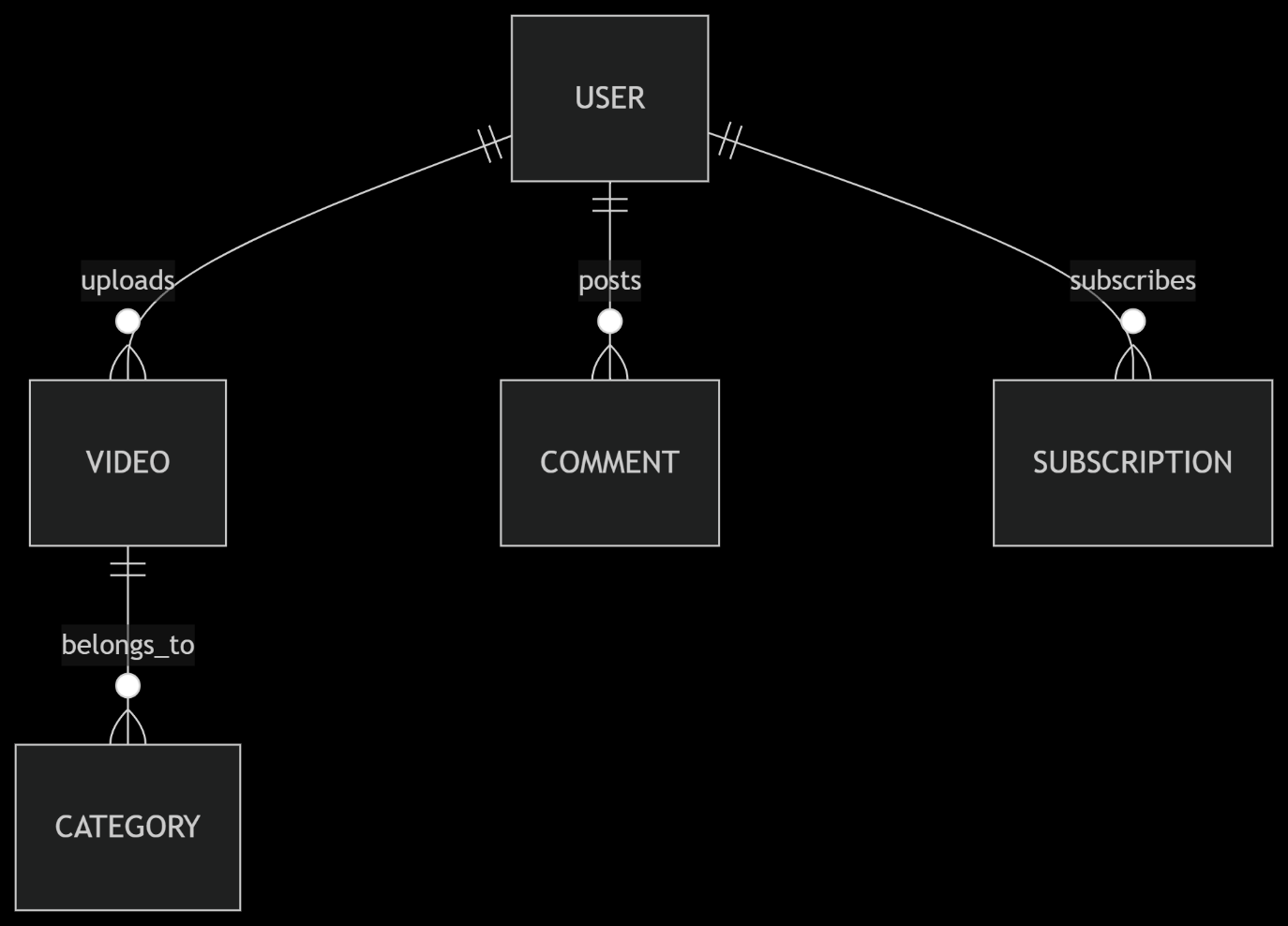
Code

Download

Mermaid rendering failed.

**8. Appendix**

**8.1 ER Diagram**



**8.2 Sample MongoDB Schemas**

**User Schema**

{

email: String,

password: String,

username: String,

videos: [ObjectId],

subscriptions: [ObjectId]

}

**Video Schema**

{

videoId: String, // From Bunny CDN

title: String,

views: Number,

categories: [ObjectId]

}

**Approval**

| **Role** | **Name** | **Signature** | **Date** |
| --- | --- | --- | --- |
| Developer | [Your Name] |  | [DD/MM/YYYY] |

**Notes**:

* Replace placeholder values (e.g., [Your Name]) with actual project details.
* Refer to the video transcript for implementation specifics.
* Use tools like **Mermaid Live Editor** to render diagrams.

This SRS aligns with the demo shown in the tutorial and provides a blueprint for your project. Let me know if you need modifications!