System Requirements

Netflix-Style Video Player - System Requirements & Specifications

This document outlines the complete system requirements for running the Netflix-style Video Player application in different environments.

Table of Contents

Server Requirements

Minimum Requirements

Component	Specification
CPU	2 cores, 2.0 GHz
RAM	2 GB
Storage	10 GB free space (excluding video content)
OS	Linux (Ubuntu 18.04+), Windows 10+, macOS 10.14+
Network	10 Mbps upload bandwidth

Recommended Requirements

Component	Specification
CPU	4+ cores, 3.0+ GHz (Intel i5/AMD Ryzen 5 or better)
RAM	8+ GB
Storage	50+ GB SSD (excluding video content)
OS	Linux (Ubuntu 20.04+ LTS), Windows 11, macOS 12+
Network	100+ Mbps upload bandwidth

High-Performance Requirements (100+ Users)

Component	Specification
CPU	8+ cores, 3.5+ GHz (Intel i7/AMD Ryzen 7 or better)
RAM	16+ GB
Storage	100+ GB NVMe SSD + separate storage for videos
OS	Linux (Ubuntu 22.04 LTS recommended)
Network	1+ Gbps dedicated bandwidth

Client Requirements

Desktop/Laptop

Component	Minimum	Recommended
CPU	Dual-core 1.8 GHz	Quad-core 2.5+ GHz

Component	Minimum	Recommended
RAM	4 GB	8+ GB
GPU	Integrated graphics	Dedicated GPU (for 4K content)
Storage	1 GB free space	5+ GB free space
Network	5 Mbps download	25+ Mbps download

Mobile Devices

Component	Minimum	Recommended
RAM	2 GB	4+ GB
Storage	500 MB free	2+ GB free
Network	3 Mbps download	10+ Mbps download
OS	Android 7.0+, iOS 12+	Android 10+, iOS 14+

Smart TV / Set-Top Box

Component	Minimum	Recommended
RAM	1 GB	2+ GB
Storage	500 MB free	1+ GB free
Network	10 Mbps download	25+ Mbps download
OS	Android TV 7.0+, webOS 4.0+	Android TV 10+, webOS 6.0+

Software Dependencies

Core Runtime Dependencies

Software	Version	Purpose
Node.js	18.0.0+	JavaScript runtime
NPM	8.0.0+	Package manager
FFmpeg	4.4.0+	Video processing & transcoding
SQLite3	3.36.0+	Database engine

Required Node.js Packages

```
{
  "express": "^4.18.0",
  "cors": "^2.8.5",
  "cookie-parser": "^1.4.6",
  "sqlite3": "^5.1.6",
  "fluent-ffmpeg": "^2.1.2",
  "multer": "^1.4.5",
  "dotenv": "^16.0.3"
}
```

Optional Dependencies

Software	Version	Purpose
PM2	5.0.0+	Production process management
Nginx	1.18.0+	Reverse proxy & load balancing
Docker	20.0.0+	Containerization
Redis	6.0.0+	Caching (future enhancement)

Hardware Recommendations

Development Environment

```
CPU: Intel i5-8400 / AMD Ryzen 5 3600
RAM: 8 GB DDR4
Storage: 256 GB SSD
Network: 50 Mbps broadband
```

Small Production (1-20 Users)

```
CPU: Intel i5-10400 / AMD Ryzen 5 5600
RAM: 16 GB DDR4
Storage: 500 GB NVMe SSD + 2 TB HDD for videos
Network: 100 Mbps dedicated
```

Medium Production (20-100 Users)

```
CPU: Intel i7-11700 / AMD Ryzen 7 5700X
RAM: 32 GB DDR4
Storage: 1 TB NVMe SSD + 4 TB HDD RAID 1
Network: 500 Mbps dedicated
GPU: Optional for hardware transcoding
```

Large Production (100+ Users)

```
CPU: Intel Xeon E-2288G / AMD EPYC 7302P
RAM: 64 GB DDR4 ECC
Storage: 2 TB NVMe SSD + 8 TB HDD RAID 5
Network: 1 Gbps dedicated
GPU: NVIDIA Tesla T4 (for hardware transcoding)
Load Balancer: Multiple server instances
```

Network Requirements

Bandwidth Requirements per Stream Quality

Quality	Resolution	Bitrate	Bandwidth per User
Low	480p	1 Mbps	1.2 Mbps
Medium	720p	3 Mbps	3.5 Mbps

Quality	Resolution	Bitrate	Bandwidth per User
High	1080p	5 Mbps	6 Mbps
Ultra	4K	15 Mbps	18 Mbps

Server Upload Requirements

```
Formula: (Number of Users × Average Quality Bitrate) × 1.2
(overhead)

Examples:
- 10 users @ 1080p: 10 × 6 Mbps × 1.2 = 72 Mbps
- 50 users @ 720p: 50 × 3.5 Mbps × 1.2 = 210 Mbps
- 100 users @ mixed quality: ~400 Mbps recommended
```

Network Configuration

- **Latency**: < 50ms for optimal experience
- **Packet Loss**: < 0.1%
- **Jitter**: < 10ms
- **Port Requirements**:
  - HTTP: 80, 443 (if using HTTPS)
  - Custom: 5555 (default application port)

Storage Requirements

Application Files

- **Base Installation**: 500 MB
- **Node Modules**: 200 MB
- **Logs**: 100 MB (rotating)
- **Thumbnails Cache**: 1-5 GB (depends on video library)
- **Database**: 10-100 MB (depends on users/metadata)

Video Content Storage

Library Size	Storage Required	RAID Recommendation
Small (< 500 videos)	1-2 TB	Single drive + backup

Library Size	Storage Required	RAID Recommendation
Medium (500-2000 videos)	5-10 TB	RAID 1 (mirroring)
Large (2000+ videos)	20+ TB	RAID 5/6 with hot spare

Storage Performance

- **SSD Required**: For application, database, and thumbnail cache
- **HDD Acceptable**: For video content storage
- **Read Speed**: Minimum 100 MB/s for video streaming
- **Write Speed**: Minimum 50 MB/s for thumbnail generation

Browser Support

Desktop Browsers

Browser	Minimum Version	Recommended Version	Features
Chrome	90+	Latest	Full support
Firefox	88+	Latest	Full support
Safari	14+	Latest	Full support
Edge	90+	Latest	Full support
Opera	76+	Latest	Full support

Mobile Browsers

Browser	Minimum Version	Features
Chrome Mobile	90+	Full support
Safari iOS	14+	Full support
Firefox Mobile	88+	Full support
Samsung Internet	14+	Full support

Smart TV Browsers

Platform	Support Level	Notes
Android TV	Full	Chrome-based browser
webOS (LG)	Full	Built-in browser
Tizen (Samsung)	Full	Built-in browser
Fire TV	⚠ Limited	Silk browser
Roku	No	No web browser support

Development Environment

Required Software

```
# Node.js and NPM
Node.js 18.0.0+
NPM 8.0.0+

# FFmpeg
FFmpeg 4.4.0+ with libx264, libx265, libvpx

# Database
```

```
SQLite3 3.36.0+

# Version Control
Git 2.30.0+

# Code Editor (recommended)
VS Code with extensions:
- Node.js Extension Pack
- SQLite Viewer
- REST Client
```

Development Setup Commands

```
# Install Node.js dependencies
npm install

# Install FFmpeg (Ubuntu/Debian)
sudo apt update
sudo apt install ffmpeg

# Install FFmpeg (macOS)
brew install ffmpeg

# Install FFmpeg (Windows)
# Download from https://ffmpeg.org/download.html

# Verify installation
node --version    # Should be 18+
npm --version     # Should be 8+
ffmpeg -version   # Should be 4.4+
```

Production Environment

Linux Server Setup (Ubuntu 22.04 LTS)

```
# System updates
sudo apt update && sudo apt upgrade -y

# Install Node.js 18
curl -fsSL https://deb.nodesource.com/setup_18.x | sudo -E bash -
sudo apt-get install -y nodejs

# Install FFmpeg
sudo apt install ffmpeg -y
```

```
# Install PM2 globally
sudo npm install -g pm2

# Create application user
sudo adduser --system --group videostreamer

# Set up directories
sudo mkdir -p /opt/video-player
sudo mkdir -p /var/log/video-player
sudo chown -R videostreamer:videostreamer /opt/video-player
sudo chown -R videostreamer:videostreamer /var/log/video-player
```

PM2 Production Configuration

```
// ecosystem.config.js
module.exports = {
  apps: [{
    name: 'video-player',
    script: 'server.js',
    instances: 'max',
    exec_mode: 'cluster',
    env: {
      NODE_ENV: 'production',
      PORT: 5555
    },
    log_file: '/var/log/video-player/combined.log',
    out_file: '/var/log/video-player/out.log',
    error_file: '/var/log/video-player/error.log',
    max_memory_restart: '1G',
    watch: false,
    ignore_watch: ['node_modules', 'videos', 'logs']
  }]
};
```

Nginx Reverse Proxy Configuration

```
server {
    listen 80;
    server_name your-domain.com;

    client_max_body_size 100M;

    location / {
        proxy_pass http://localhost:5555;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For
$proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
        proxy_cache_bypass $http_upgrade;

        # Video streaming optimizations
        proxy_buffering off;
        proxy_request_buffering off;
    }
}
```

Performance Benchmarks

Expected Performance Metrics

Server Performance

Metric	Target	Good	Excellent
Response Time	< 200ms	< 100ms	< 50ms
Video Start Time	< 3s	< 2s	< 1s
Thumbnail Load	< 1s	< 500ms	< 200ms
CPU Usage	< 80%	< 60%	< 40%
Memory Usage	< 80%	< 60%	< 40%

Client Performance

Metric	Target	Good	Excellent
Page Load Time	< 3s	< 2s	< 1s
Video Buffer Time	< 5s	< 3s	< 1s
UI Responsiveness	< 100ms	< 50ms	< 20ms

Load Testing Results

```
Tested Configuration:
- Server: 4-core, 8GB RAM, SSD
- Network: 100 Mbps
- Video Quality: 1080p

Results:
- 10 concurrent users:   Excellent (< 1s start time)
- 25 concurrent users:   Good (< 2s start time)
- 50 concurrent users: ⚠ Acceptable (< 3s start time)
- 100+ concurrent users:   Requires scaling
```

Scaling Considerations

Horizontal Scaling Options

Load Balancer Setup

```
[Internet] → [Load Balancer] → [App Server 1]
                            → [App Server 2]
                            → [App Server 3]
                            → [Shared Storage]
```

Database Scaling

- **Read Replicas**: For user data and metadata
- **Database Sharding**: By user ID or content type
- **Caching Layer**: Redis for session and metadata caching

CDN Integration

```
[Users] → [CDN Edge] → [Origin Server]
                   → [Video Storage]
```

Vertical Scaling Limits

Component	Scale-up Limit	Recommendation
CPU	32+ cores	Switch to horizontal scaling
RAM	128+ GB	Consider clustering
Storage	100+ TB	Implement distributed storage
Network	10+ Gbps	Use CDN for video delivery

Security Requirements

Server Security

- **OS Updates**: Regular security patches
- **Firewall**: Block unnecessary ports
- **SSL/TLS**: HTTPS for all connections
- **User Permissions**: Non-root application user
- **File Permissions**: Restricted access to video files

Application Security

- **Input Validation**: All user inputs sanitized
- **SQL Injection**: Parameterized queries
- **XSS Protection**: Content Security Policy
- **CSRF Protection**: Token-based validation
- **Rate Limiting**: API endpoint protection

Network Security

- **VPN Access**: For administrative functions
- **DDoS Protection**: Cloudflare or similar
- **Intrusion Detection**: Fail2ban or equivalent
- **Log Monitoring**: Centralized logging system

Docker Requirements

Docker Compose Configuration

```yaml
version: '3.8'
services:
  video-player:
    build: .
    ports:
      - "5555:5555"
    volumes:
      - ./videos:/app/videos
      - ./databases:/app/databases
    environment:
      - NODE_ENV=production
      - PORT=5555
    restart: unless-stopped

  nginx:
    image: nginx:alpine
    ports:
      - "80:80"
      - "443:443"
    volumes:
      - ./nginx.conf:/etc/nginx/nginx.conf
    depends_on:
      - video-player
    restart: unless-stopped
```

Container Requirements

- **Base Image**: node:18-alpine
- **Memory Limit**: 1-4 GB per container
- **CPU Limit**: 1-4 cores per container
- **Volume Mounts**: Videos, databases, logs
- **Health Checks**: HTTP endpoint monitoring

System Verification Checklist

Pre-Installation Checklist

- □ Server meets minimum hardware requirements
- □ Operating system is supported and updated
- □ Network bandwidth is adequate
- □ Storage space is sufficient

- □ Backup strategy is in place

Post-Installation Verification

```
# Check Node.js version
node --version

# Check FFmpeg installation
ffmpeg -version

# Test video streaming
curl -I http://localhost:5555/api/video/test/sample.mp4

# Check system resources
htop
df -h
free -h

# Test multi-user load
# Use Postman collection for API testing
```

Performance Monitoring

- **System Metrics**: CPU, RAM, Disk I/O, Network
- **Application Metrics**: Response times, error rates
- **User Experience**: Video start times, buffering events
- **Resource Usage**: FFmpeg processes, database queries

Support & Troubleshooting

Common Issues

1. **Video won't play**: Check FFmpeg installation and codec support
2. **Slow loading**: Verify network bandwidth and server resources
3. **High CPU usage**: Consider hardware transcoding or quality reduction
4. **Database errors**: Check SQLite file permissions and disk space

Monitoring Tools

- **System**: htop, iotop, netstat
- **Application**: PM2 monit, custom logging
- **Network**: iftop, tcpdump
- **Performance**: New Relic, DataDog (optional)

Log Locations

```
Application Logs: /var/log/video-player/
PM2 Logs: ~/.pm2/logs/
System Logs: /var/log/syslog
Nginx Logs: /var/log/nginx/
```

Version History

Version	Date	Changes
1.0.0	2025-09-20	Initial system requirements
1.1.0	Future	Theme system requirements
1.2.0	Future	Scaling and CDN integration

Note: These requirements are based on the current feature set. Future updates may require additional resources or dependencies.

Last Updated: September 2025