

# MVP Task Platform - Setup Guide

## Prerequisites

- Node.js (v16+)
- npm or yarn
- Git

## Quick Setup

### 1. Clone & Install

```
git clone  
cd mvp_task_platform
```

### 2. Backend Setup

```
cd backend  
npm install  
npm install --save-dev jest supertest  
cp .env.example .env  
npm run dev
```

### 3. Frontend Setup (new terminal)

```
cd frontend  
npm install  
npm start
```

### 4. Access Application

- Frontend: <http://localhost:3000>
- Backend: <http://localhost:5002>

# Configuration

## Environment Variables (backend/.env)

```
PORT=5002
JWT_SECRET=your-secret-key
JWT_EXPIRE=7d
EMAIL_HOST=smtp.gmail.com
EMAIL_PORT=587
EMAIL_USER=your-email@gmail.com
EMAIL_PASS=your-app-password
```

## Database

- SQLite (auto-created)
- Location: `backend/src/db/database.sqlite`

## First User

- Go to `http://localhost:3000`
- Register an account
- First user becomes admin automatically

# Development

## Backend Commands

```
npm run dev      # Development server
npm start        # Production server
npm test         # Run Jest tests
```

## Frontend Commands

```
npm start        # Development server
npm run build   # Production build
npm test         # Run Jest tests
```

# Testing

**Backend:** `cd backend && npm install --save-dev jest && npm test`

**Frontend:** `cd frontend && npm test`

# Troubleshooting

## Port Issues

```
lsof -i:5002      # Find process using port  
kill -9          # Kill process
```

## Common Fixes

- ▶ Ensure backend runs before frontend
- ▶ Check `.env` file configuration
- ▶ Reset database: delete `backend/src/db/database.sqlite`
- ▶ Check console for error messages

# Project Structure

```
mvp_task_platform/  
|   └── backend/src/  
|       |   └── routes/      # API endpoints  
|       |   └── models/      # Database models  
|       |   └── middleware/  # Auth & validation  
|       └── controllers/   # Business logic  
|   └── frontend/src/  
|       |   └── pages/       # Main pages  
|       |   └── components/  # UI components  
|       |   └── context/     # State management  
|       └── utils/         # API client
```

# Features

- User authentication & authorization
- Task management (CRUD operations)
- Real-time collaboration with comments
- Activity tracking
- Admin dashboard
- Dark/Light theme
- Responsive design

## Usage

- Register/Login at <http://localhost:3000>
- Create and manage tasks
- Collaborate with comments
- Track team activity
- Admin users can manage platform

---

**Last Updated:** December 2025

**Created by:** Anuja S Nair