MERN MySQL Full Stack Application

A complete full-stack application built with MySQL, Express, React (with Vite), and Node.js.

🚀 Tech Stack

Backend

- **Node.js** - Runtime environment
- **Express** - Web framework
- **MySQL2** - Database
- **JWT** - Authentication
- **Bcrypt** - Password hashing
- **dotenv** - Environment variables
- **CORS** - Cross-origin resource sharing
- **Helmet** - Security middleware
- **Morgan** - HTTP request logger
- **Compression** - Response compression

Frontend

- **React 18** - UI library
- **Vite** - Build tool
- **React Router v6** - Routing
- **TanStack Query (React Query)** - Data fetching & caching
- **Axios** - HTTP client
- **React Hot Toast** - Notifications

📁 Project Structure

```
mern-mysql-app/
├── backend/
│   ├── config/
│   │   └── database.js
│   ├── controllers/
│   │   └── userController.js
│   ├── models/
│   │   └── User.js
│   ├── routes/
│   │   └── userRoutes.js
│   ├── .env.example
```

```
│   ├── .gitignore
│   ├── package.json
│   └── server.js
├── frontend/
│   ├── src/
│   │   ├── config/
│   │   │   └── api.js
│   │   ├── hooks/
│   │   │   └── useUserQueries.js
│   │   ├── pages/
│   │   │   ├── Home.jsx
│   │   │   ├── Users.jsx
│   │   │   ├── Login.jsx
│   │   │   └── Register.jsx
│   │   ├── services/
│   │   │   └── userService.js
│   │   ├── App.jsx
│   │   ├── main.jsx
│   │   └── index.css
│   ├── .env.example
│   ├── .gitignore
│   ├── index.html
│   ├── package.json
│   └── vite.config.js
└── README.md
```

🛠️ Local Development Setup

Prerequisites

- Node.js (v18 or higher)
- MySQL (v8 or higher)
- Git

1. Clone the Repository

```bash
git clone <your-repo-url>
cd mern-mysql-app
```

2. Database Setup

Create a MySQL database:

```sql
```

```
CREATE DATABASE mern_app_db;
```

The users table will be created automatically when you start the backend server.

3. Backend Setup

```bash
cd backend
npm install

# Create .env file
cp .env.example .env
```

Edit `.env` file with your local MySQL credentials:

```env
NODE_ENV=development
PORT=5000
DB_HOST=localhost
DB_USER=your_mysql_username
DB_PASSWORD=your_mysql_password
DB_NAME=mern_app_db
DB_PORT=3306
JWT_SECRET=your_secret_key_here
JWT_EXPIRE=7d
CORS_ORIGIN=http://localhost:5173
API_VERSION=v1
```

Start the backend server:

```bash
npm run dev
```

Backend will run on `http://localhost:5000`

4. Frontend Setup

```bash
cd frontend
npm install

# Create .env file
cp .env.example .env
```

Edit `.env` file:

```env
VITE_API_URL=http://localhost:5000/api/v1
```

Start the frontend development server:

```bash
npm run dev
```

Frontend will run on `http://localhost:5173`

🌐 VPS Deployment (Hostinger)

Prerequisites on VPS

- Ubuntu 20.04 or higher
- Node.js and npm
- MySQL
- Nginx (for reverse proxy)
- PM2 (for process management)
- Git

1. Connect to Your VPS

```bash
ssh root@your_vps_ip
```

2. Install Required Software

```bash
```

```bash
# Update system
apt update && apt upgrade -y

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

# Install MySQL
apt install mysql-server -y

# Install Nginx
apt install nginx -y

# Install PM2 globally
npm install -g pm2

# Install Git
apt install git -y
```

3. Setup MySQL on VPS

```bash
# Secure MySQL installation
mysql_secure_installation

# Login to MySQL
mysql -u root -p

# Create database and user
CREATE DATABASE mern_app_db;
CREATE USER 'mern_user'@'localhost' IDENTIFIED BY 'strong_password_here';
GRANT ALL PRIVILEGES ON mern_app_db.* TO 'mern_user'@'localhost';
FLUSH PRIVILEGES;
EXIT;
```

4. Clone Your Repository on VPS

```bash
cd /var/www
git clone <your-github-repo-url>
cd mern-mysql-app
```

5. Backend Setup on VPS

```bash
cd backend
npm install --production

# Create .env file
nano .env
```

Add production environment variables:

```env
NODE_ENV=production
PORT=5000
DB_HOST=localhost
DB_USER=mern_user
DB_PASSWORD=strong_password_here
DB_NAME=mern_app_db
DB_PORT=3306
JWT_SECRET=production_jwt_secret_key
JWT_EXPIRE=7d
CORS_ORIGIN=https://yourdomain.com
API_VERSION=v1
```

Start backend with PM2:

```bash
pm2 start server.js --name mern-backend
pm2 save
pm2 startup
```

6. Frontend Build and Setup

```bash
cd /var/www/mern-mysql-app/frontend

# Create .env.production file
nano .env
```

Add production API URL:

```env
```

```
VITE_API_URL=https://yourdomain.com/api/v1
```

Build the frontend:

```bash
npm install
npm run build
```

7. Configure Nginx

Create Nginx configuration:

```bash
nano /etc/nginx/sites-available/mern-app
```

Add the following configuration:

```nginx
server {
    listen 80;
    server_name yourdomain.com www.yourdomain.com;

    # Frontend
    location / {
        root /var/www/mern-mysql-app/frontend/dist;
        try_files $uri $uri/ /index.html;
    }

    # Backend API
    location /api {
        proxy_pass http://localhost:5000;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
        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;
    }
}
```

Enable the site:

```bash
ln -s /etc/nginx/sites-available/mern-app /etc/nginx/sites-enabled/
nginx -t
systemctl restart nginx
```

8. Setup SSL with Let's Encrypt (Optional but Recommended)

```bash
apt install certbot python3-certbot-nginx -y
certbot --nginx -d yourdomain.com -d www.yourdomain.com
```

🔄 Automatic Deployment with GitHub Actions

Create `.github/workflows/deploy.yml` in your repository:

```yaml
```

```yaml
name: Deploy to VPS

on:
  push:
    branches: [ main ]

jobs:
  deploy:
    runs-on: ubuntu-latest

    steps:
    - name: Deploy to VPS
      uses: appleboy/ssh-action@master
      with:
        host: ${{ secrets.VPS_HOST }}
        username: ${{ secrets.VPS_USERNAME }}
        key: ${{ secrets.VPS_SSH_KEY }}
        script: |
          cd /var/www/mern-mysql-app
          git pull origin main

          # Backend
          cd backend
          npm install --production
          pm2 restart mern-backend

          # Frontend
          cd ../frontend
          npm install
          npm run build

          # Restart Nginx
          sudo systemctl restart nginx
```

Setup GitHub Secrets

1. Go to your GitHub repository
2. Settings → Secrets and variables → Actions
3. Add these secrets:
   - `VPS_HOST`: Your VPS IP address
   - `VPS_USERNAME`: SSH username (usually `root`)
   - `VPS_SSH_KEY`: Your private SSH key

Generate SSH Key for GitHub Actions

On your local machine:

```bash
ssh-keygen -t rsa -b 4096 -C "github-actions"
```

Copy the public key to your VPS:

```bash
ssh-copy-id -i ~/.ssh/id_rsa.pub root@your_vps_ip
```

Copy the private key content and add it to GitHub Secrets as `VPS_SSH_KEY`.

🔧 Alternative: Manual Deployment Script

Create `deploy.sh` on your VPS:

```bash
#!/bin/bash
cd /var/www/mern-mysql-app

echo "Pulling latest changes..."
git pull origin main

echo "Updating backend..."
cd backend
npm install --production
pm2 restart mern-backend

echo "Building frontend..."
cd ../frontend
npm install
npm run build

echo "Restarting Nginx..."
sudo systemctl restart nginx

echo "Deployment completed!"
```

Make it executable:

```bash
chmod +x deploy.sh
```

Run deployment:

```bash
./deploy.sh
```

📝 API Endpoints

Users

- `GET /api/v1/users` - Get all users
- `GET /api/v1/users/:id` - Get single user
- `POST /api/v1/users/register` - Register new user
- `POST /api/v1/users/login` - Login user
- `PUT /api/v1/users/:id` - Update user
- `DELETE /api/v1/users/:id` - Delete user

🐛 Troubleshooting

Backend won't start

- Check MySQL connection
- Verify .env variables
- Check if port 5000 is available

Frontend can't connect to API

- Verify VITE_API_URL in .env
- Check CORS settings in backend
- Ensure backend is running

PM2 issues

```bash
pm2 logs mern-backend
pm2 restart mern-backend
```

Nginx issues

```bash
nginx -t
systemctl status nginx
journalctl -xe
```

📄 License

MIT

👨‍💻 Author

Your Name

> This README provides comprehensive documentation for local development and VPS deployment!