# الواجهة الخلفية

## ملخص

تم استخدام إطار العمل Node Js و مكتبة Express لإنشاء api لتطبيق المكتبة الالكترونية، وبناء قاعدة بيانات mysql.

# التشغيل والاختبار

## قاعدة البيانات

تم استخدام قاعدة البيانات mysql، لذاك جيب انشاء قاعدة بيانات على مخدم مثل XAMPP وإنشاء الجداول الخاصة بها عن طريق التعليمات التالية:

|  |
| --- |
| book CREATE TABLE `book` (   `Id` int(11) NOT NULL AUTO\_INCREMENT,   `Title` varchar(100) DEFAULT NULL,   `Type` varchar(50) DEFAULT NULL,   `Price` decimal(10,2) DEFAULT NULL,   `pubId` int(11) DEFAULT NULL,   `AuthorId` int(11) DEFAULT NULL,   PRIMARY KEY (`Id`),   KEY `pubId` (`pubId`),   KEY `AuthorId` (`AuthorId`),   CONSTRAINT `book\_ibfk\_1` FOREIGN KEY (`pubId`) REFERENCES `publisher` (`Id`),   CONSTRAINT `book\_ibfk\_2` FOREIGN KEY (`AuthorId`) REFERENCES `author` (`Id`)  ); |
| author CREATE TABLE `author` (   `Id` int(11) NOT NULL AUTO\_INCREMENT,   `Fname` varchar(50) DEFAULT NULL,   `Lname` varchar(50) DEFAULT NULL,   `Country` varchar(50) DEFAULT NULL,   `City` varchar(50) DEFAULT NULL,   `Address` varchar(255) DEFAULT NULL,   PRIMARY KEY (`Id`)  ); |
| user CREATE TABLE `user` (   `Id` int(11) NOT NULL AUTO\_INCREMENT,   `Username` varchar(50) NOT NULL,   `Password` varchar(255) NOT NULL,   `FName` varchar(50) DEFAULT NULL,   `LName` varchar(50) DEFAULT NULL,   PRIMARY KEY (`Id`),   UNIQUE KEY `Username` (`Username`)  ); |
| publisher CREATE TABLE `publisher` (   `Id` int(11) NOT NULL AUTO\_INCREMENT,   `PName` varchar(100) DEFAULT NULL,   `City` varchar(50) DEFAULT NULL,   PRIMARY KEY (`Id`)  ); |

يتم إدراج معلومات الاتصال بقاعدة البيانات في ملف `.env` مثل اسم المستخدم المحلي وكلمة المرور الخاصة به وأخيرا اسم قاعدة البيانات المنشئة.

على الشكل التالي:

|  |
| --- |
| DB\_HOST=localhost  DB\_USER=sliman  DB\_PASSWORD=123456  DB\_NAME=e\_library |

قبل تشغيل المشؤوع نقوم بتحميل المتطلبات عن طريق الانتقال الى المجلد الحاوي للمشروع ثم الأمر التالي:

npm install

يمكن تشغيل المشروع من cmd

node server.js

# هيكل المشروع

|  |
| --- |
|  |

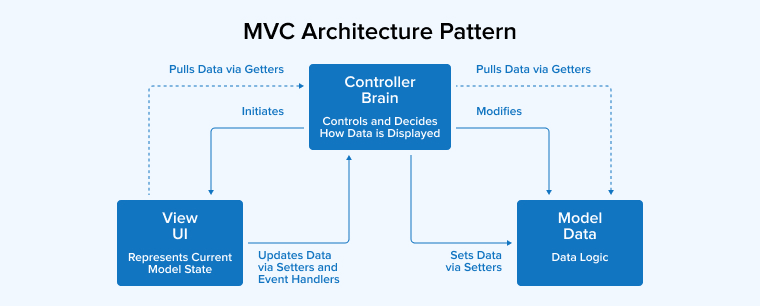
## نظرة عامة على المشروع باستخدام Node.js و Express

المشروع يعتمد على **Node**.**js** مع **Express** لبناء **واجهة برمجية (API Backend)** لإدارة البيانات والتفاعل مع قاعدة بيانات، غالبًا باستخدام **Sequelize** كـ ORM. الفكرة الأساسية هي أن Express يوفر بنية قوية لإنشاء **مسارات (Routes)** لمعالجة طلبات HTTP، ودمجها مع قاعدة بيانات لإدارة البيانات.

## كيف يعمل Node.js و Express؟

- **Node.js**: بيئة تشغيل JavaScript خارج المتصفح تستخدم نظام "Single-threaded, Non-blocking I/O" مما يجعلها مثالية لإنشاء تطبيقات ذات أداء عالي تعتمد على الأحداث.

- **Express.js**: إطار عمل خفيف مبني فوق Node.js يساعد في تنظيم التطبيق عبر المسارات (Routes) والـ Middleware.



1MVC Architecture

## ملفات المشروع الأساسية

في المشروع، لدينا عدة ملفات منظمة وفقًا للوظائف الأساسية:

### 1️⃣ ملف `server.js` (نقطة البداية للتطبيق)

* يقوم بإعداد `Express`
* يربط قاعدة البيانات
* يُعرّف المسارات الأساسية
* يشغل السيرفر على المنفذ المحدد

### 2️⃣ ملفات النماذج (`models/`) (تمثيل الجداول داخل قاعدة البيانات)

- تستخدم **Sequelize** لإنشاء نماذج مثل `Book.js`, `Author.js`, `Publisher.js`

- تحتوي كل نموذج على تعريف هيكله (الأعمدة والعلاقات)

- توفر طريقة لربط الجداول ببعضها

### 3️⃣ ملفات المسارات (`routes/`) (تنظيم نقاط الوصول)

- كل ملف يحتوي على مسارات لمعالجة طلبات المستخدمين

- يستخدم `express.Router()` لفصل إدارة المسارات بطريقة منظمة

- يتم ربط المسارات بـ `server.js`

### 4️⃣ ملف قاعدة البيانات (`config/db.js`) (إدارة اتصال قاعدة البيانات)

- يحتوي على إعدادات الاتصال بـ MySQL

### الفكرة العامة للمشروع باختصار

- **Node.js** مع **Express** لإنشاء الـ **Backend**

- **Sequelize** لإدارة قاعدة البيانات والتعامل مع الجداول

- نظام نماذج (`models/`) يحتوي على تعريف الجداول والعلاقات

- نظام مسارات (`routes/`) لمعالجة طلبات المستخدمين وجلب البيانات

- تهيئة قاعدة بيانات (`config/db.js`) لربط المشروع مع **MySQL**