1. **Setup And Creating HR in Oracle**

 **Install Oracle Database**:

* Download and install [Oracle Database](https://www.oracle.com/database/) on your local machine. You can use Oracle Express Edition (XE) if you’re looking for a lightweight option.
* Set up the environment variables as instructed during installation.

 **Set Up SQL Developer**:

* Download and install [Oracle SQL Developer](https://www.oracle.com/tools/sql-developer.html).
* Use it to connect to your Oracle Database.
* Import the hr.sql file (if provided) using the SQL Developer interface or command-line tools:

@C:\path\to\hr.sql

 **Verify Data**:

* Check that tables like Books, Authors, Customers, and Purchases exist and are populated.

1. **Creating rest api with Node js**

I used **Node.js** with express and oracledb to build the backend. Let’s start by creating basic endpoints for one entity, such as **Books**.

**1. Install Required Packages**

* Install Node.js : [Node.js Official Website](https://nodejs.org/).
* Create a new project:

mkdir book\_store\_api

cd book\_store\_api

npm init -y

* Install dependencies:

npm install express body-parser oracledb cors

1. **Testing And PostMan Documentation for api testing**

**Base URL**

http://localhost:5000

**1. Get All Authors**

* **Method**: GET
* **Endpoint**: /authors

**Request**

* No body required.

**Response**

[

[1, "Author Name", "1234567890"],

[2, "Another Author", "0987654321"]

]

**2. Add an Author**

* **Method**: POST
* **Endpoint**: /authors

**Headers**

* Content-Type: application/json

**Request Body**

{

"AuthorID": 1,

"Name": "Author Name",

"Contact": "1234567890"

}

**Response**

* **Success**: "Author added successfully"
* **Error**: Error adding author

**3. Get All Books**

* **Method**: GET
* **Endpoint**: /books

**Request**

* No body required.

**Response**

[

[1, "Book Title", "Genre", "2024-01-01", 10, 1],

[2, "Another Book", "Genre", "2023-12-25", 5, 2]

]

**4. Add a Book**

* **Method**: POST
* **Endpoint**: /books

**Headers**

* Content-Type: application/json

**Request Body**

{

"BookID": 1,

"Title": "Book Title",

"Genre": "Genre",

"PublicationDate": "2024-01-01",

"StockQuantity": 10,

"AuthorName": "Author Name",

"AuthorContact": "1234567890"

}

**Response**

* **Success**: "Book and author added successfully"
* **Error**: Error details: <error\_message>

**5. Get All Customers**

* **Method**: GET
* **Endpoint**: /customers

**Request**

* No body required.

**Response**

[

[1, "Customer Name", "email@example.com", "1234567890"],

[2, "Another Customer", "another@example.com", "0987654321"]

]

**6. Add a Customer**

* **Method**: POST
* **Endpoint**: /customers

**Headers**

* Content-Type: application/json

**Request Body**

{

"CustomerID": 1,

"Name": "Customer Name",

"Email": "email@example.com",

"Phone": "1234567890"

}

**Response**

* **Success**: "Customer added successfully"
* **Error**: Error adding customer

**7. Get All Purchases**

* **Method**: GET
* **Endpoint**: /purchases

**Request**

* No body required.

**Response**

[

[1, 1, "2024-01-01"],

[2, 2, "2024-02-01"]

]

**8. Add a Purchase**

* **Method**: POST
* **Endpoint**: /purchases

**Headers**

* Content-Type: application/json

**Request Body**

{

"PurchaseID": 1,

"CustomerID": 1,

"PurchaseDate": "2024-01-01"

}

**Response**

* **Success**: "Purchase added successfully"
* **Error**: Error adding purchase

**9. Get All Purchase Items**

* **Method**: GET
* **Endpoint**: /purchase-items

**Request**

* No body required.

**Response**

[

[1, 1, 1],

[2, 1, 2]

]

**10. Add a Purchase Item**

* **Method**: POST
* **Endpoint**: /purchase-items

**Headers**

* Content-Type: application/json

**Request Body**

{

"PurchaseItemID": 1,

"PurchaseID": 1,

"BookID": 1

}

**Response**

* **Success**: "Purchase item added successfully"
* **Error**: Error adding purchase item