Questions

1. What would be the main API endpoints you would create when building this solution?

HTTP Method	URL	Description
GET	/api/v1/users	Find all Users
GET	/api/v1/users/{id}	Find Users by id
POST	/api/v1/users	Create new users
POST	/api/v1/users/authenticate	Authenticate the users
PUT	/api/v1/users/{id}	Update users by id
DELETE	/api/v1/users/{id}	Delete users by id(permanent)
GET	/api/v1/medication	Find all medication
GET	/api/v1/medication/{id}	Find medication by id
POST	/api/v1/medication	Create new medication
PUT	/api/v1/medication	Update medication by id
DELETE	/api/v1/medication/{id}	Delete medication
GET	/api/v1/customers	Find all customers
GET	/api/v1/customers/{id}	Find customer by id
POST	/api/v1/customers	Create new customer
PUT	/api/v1/customers/{id}	Update customer record by id
Delete	/api/v1/customers/{id}	Delete customer by id

2. Provide an ER Diagram or Class / Interface / DDL /

```
CREATE TABLE users (
  id INT NOT NULL AUTO_INCREMENT,
  name VARCHAR(255) NOT NULL,
  username VARCHAR(255) NOT NULL UNIQUE,
  password VARCHAR(255) NOT NULL,
  role VARCHAR(255) NOT NULL,
  PRIMARY KEY (id)
);


CREATE TABLE medications (
  id INT NOT NULL AUTO_INCREMENT,
  name VARCHAR(255) NOT NULL,
  description TEXT NOT NULL,
  quantity INT NOT NULL,
  created_at DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
  updated_at DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
  deleted_at DATETIME,
  PRIMARY KEY (id)
);

CREATE TABLE customers (
  id INT NOT NULL AUTO_INCREMENT,
  name VARCHAR(255) NOT NULL UNIQUE,
  created_at DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
  updated_at DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
  deleted_at DATETIME,
  PRIMARY KEY (id)
);
```

3. What are the frameworks / libraries you would use to build this solution?

Name of the package	What would you accomplish using that
ExpressJS	Web application framework
Passport.js	User authentication
JWTs (JSON Web Tokens)	Session management
Sequelize(ORM for Node)	Database interaction and queries

4. What are the main programs and tools you would use to build this solution?

Name of programme	What would you accomplish using that?
Visual Studio Code	IDE
Node.js	Runtime environment for server side
MySQL	Database management
Postman	Check API endpoints during development

5. What is the approach you would use to implement User Permissions based on User Roles?

Assign roles to each user (Owner, Manager, and Cashier). Assign flags to customer records and medication records to indicate whether they have been soft-deleted or not, and timestamp when they were created, last modified, and soft-deleted.

Use middleware to protect API endpoints. Check the user's role and verify that the type of user is allowed for the particular endpoint.

Only allow users to access the endpoint if they have the appropriate permissions

Ex:- using Express.js for role based authrization

```
const checkRole = (requiredRole) => {
  return (req, res, next) => {
    // lets assume user role is stored in req.user.role
    if (req.user && req.user.role === ‘Manager’) {
      next(); // User has permission, continue
    } else {
res.status(403).json({ message: 'Access denied' }); // a user with this type role doesn't have permission
    }
  };
};
```

When a Manager deletes a record in Customer records or Medication records, the 'Deleted_at' column in the records gets updated with the date and time. Therefore, when displaying these records to cashiers and managers, the system will only show records with 'Deleted_at' equal to 'Null'.

However, the Owner can still view the soft-deleted records made by the Manager and has the authority to permanently delete them if desired.