**Laravel Developer Practical Exam**

**Instructions:**

* You have **2 hours** to complete this test.
* Use Laravel **10+** for development.
* Create a new Laravel project or use an existing setup.
* Ensure your code follows best practices and is well-commented.
* Submit your code in a **GitHub repository** or as a **zip file**.

**Section 1: Database Design (20 Marks)**

Design and implement the database schema for a **Student Management System** with the following tables:

1. **students** (id, name, email, created\_at, updated\_at)
2. **student\_qualifications** (id, student\_id, qualification, created\_at, updated\_at)
3. **courses** (id, name, created\_at, updated\_at)
4. **student\_courses** (id, student\_id, course\_id, created\_at, updated\_at)

**Tasks:**

* Create Laravel migrations for the above tables.
* Define **relationships** between models.
* Run the migrations and insert sample data using seeders/factories.

**Section 2: CRUD Operations (40 Marks)**

Implement CRUD functionality for managing students.

**Tasks:**

* Create a **StudentController** with methods for:
  + Listing students (index)
  + Adding a new student (create, store)
  + Editing student details (edit, update)
  + Deleting a student (destroy)
* Ensure students can be assigned **multiple courses**.
* Create **Blade views** for adding, editing, and listing students.
* Implement form validation for required fields.

**Section 3: API Development (20 Marks)**

Develop a REST API for student management.

**Tasks:**

* Create an API route group (api.php) for:
  + Listing all students (GET /api/students)
  + Retrieving a single student (GET /api/students/{id})
  + Creating a new student (POST /api/students)
  + Updating student details (PUT /api/students/{id})
  + Deleting a student (DELETE /api/students/{id})
* Return responses in **JSON format**.

**Section 4: Bonus (20 Marks)**

* Implement authentication using **Laravel Sanctum**.
* Add **search functionality** to filter students by name or email.
* Implement **AJAX** for adding and deleting courses dynamically.

**Submission Guidelines:**

* Push your code to **GitHub** and share the repository link.
* If submitting via email, compress your project into a **zip file**.
* Include a **README.md** file with setup instructions.

**Good luck!** 🚀