namespace Demo.Controllers

{

[ApiController]

[Route("api/[controller]")]

public class StudentApiController : ControllerBase

{

private readonly IMongoCollection<Student> studentCollection;

public StudentApiController()

{

string connectionString = "mongodb://localhost:27017";

var client = new MongoClient(connectionString);

var database = client.GetDatabase("YourDatabaseName");

studentCollection = database.GetCollection<Student>("Students");

}

[HttpGet] // StudentsApi

public IActionResult GetAllStudents()

{

var allStudents = studentCollection.Find(\_ => true).ToList();

return Ok(allStudents);

}

[HttpGet("{name}")] // Get by name

public IActionResult GetStudentByName(string name)

{

var student = studentCollection.Find(s => s.Name == name).FirstOrDefault();

if (student == null)

{

return NotFound($"Student with name {name} not found.");

}

return Ok(student);

}

[HttpPost]

public IActionResult AddStudent( [FromBody] CreateStudentModel newStudent)

{

var studentData = new Student

{

Name = newStudent.Name ,

Email = newStudent.Email ,

PhoneNumber = newStudent.PhoneNumber ,

Subjects = newStudent.Subjects

};

studentCollection.InsertOne( studentData ); // or newStudent van oke

return CreatedAtAction(nameof(GetStudentByName), new { name = newStudent.Name }, studentData);

}

[HttpPut("{name}")] // Update student by Name

public IActionResult UpdateStudent(string name, [FromBody] Student updatedStudent)

{

var filter = Builders<Student>.Filter.Eq(s => s.Name, name);

var updateResult = studentCollection.ReplaceOne(filter, updatedStudent);

if (updateResult.MatchedCount == 0)

{

return NotFound($"Student with name {name} not found.");

}

return NoContent();

}

// PATCH: api/StudentsApi/{name}/update-phone // Change fields dc yêu cầu

[HttpPatch("{name}/update-phone")]

public IActionResult UpdatePhoneNumber(string name, [FromBody] string newPhoneNumber)

{

var filter = Builders<Student>.Filter.Eq(s => s.Name, name);

var update = Builders<Student>.Update.Set(s => s.PhoneNumber, newPhoneNumber);

var result = studentCollection.UpdateOne(filter, update);

if (result.MatchedCount == 0)

{

return NotFound($"Student with name {name} not found.");

}

return Ok($"Phone number updated for student {name}");

}

// DELETE: api/StudentsApi/{name}

[HttpDelete("{name}")]

public IActionResult DeleteStudent(string name)

{

var deleteResult = studentCollection.DeleteOne(s => s.Name == name);

if (deleteResult.DeletedCount == 0)

{

return NotFound($"Student with name {name} not found.");

}

return Ok($"Student {name} deleted successfully.");

}

//GET /api/StudentsApi: Lấy danh sách tất cả sinh viên.

//GET /api/StudentsApi/{id}: Lấy thông tin một sinh viên theo ID.

//POST /api/StudentsApi: Tạo mới một sinh viên (kèm dữ liệu trong body).

//PUT /api/StudentsApi/{id}: Cập nhật thông tin sinh viên theo ID.

//DELETE /api/StudentsApi/{id}: Xóa sinh viên theo ID.

}

}