```java
 1 package com.infy.globaleducation.entity;
 2 import javax.persistence.Entity;
 8
 9 @Entity
10 @Table(name="student")
11 public class Student {
12 @Id
13 @GeneratedValue(strategy=GenerationType.IDENTITY)
14 private Integer studentId;
15 private String studentName;
16 private String emilId;
17 private String interestedCountry;
18 private Integer intakeYear;
19 private String studyLevel;
20 private String courseInterested;
21
22 public Integer getStudentId() {
23     return studentId;
24 }
25 public String getStudentName() {
26     return studentName;
27 }
28 public String getEmilId() {
29     return emilId;
30 }
31 public String getInterestedCountry() {
32     return interestedCountry;
33 }
34 public Integer getIntakeYear() {
35     return intakeYear;
36 }
37 public String getStudyLevel() {
38     return studyLevel;
39 }
```

```java
 1  package com.infy.globaleducation.dto;
 2  import javax.validation.constraints.Email;
 8
 9  public class StudentDTO {
10      private Integer studentId;
11
12      @NotNull(message="{Student.Name.Notpresent}")
13      //@Pattern(regexp="/^[a-z][a-z\\s]*$/",message="{Student.Name.Invalid}")
14      @Pattern(regexp = "[a-zA-Z\s]+", message="{Student.Name.Invalid}")
15      private String studentName;
16
17      @Email(message = "{Student.Email.Invalid}")
18      @NotNull(message="{Student.email.notpresent}")
19      private String emilId;
20
21      @NotNull(message="{Student.interestedCountry.notpresent}")
22      private String interestedCountry;
23
24      @NotNull(message="{Student.intakeYear.notpresent}")
25      // @Pattern(regexp = "[0-9]{4}", message="{Student.intakeyear.Invalid}")
26      private Integer intakeYear; I
27
28      @NotNull(message="{Student.studyLevel.notpresent}")
29      private String studyLevel;
30
31      @NotNull(message="{Student.courseInterested.notpresent}")
32      private String courseInterested;
33
34      public Integer getStudentId() {
35          return studentId;
36      }
37      public String getStudentName() {
38          return studentName;
39      }
40      public String getEmilId() {
```

Student.java   StudentDTO.java   StudentRepository.java ⌧

```java
1 package com.infy.globaleducation.repository;
2
3 import java.util.List;
11
12 public interface StudentRepository extends CrudRepository<Student,Integer>{
13
14
15     public Optional<Student> findByEmilId(String emilId);
16
17
18     public List<Student> findStudentByInterestedCountryAndIntakeYear(String Country, Integer intakeYear);
19 }
20
```

```java
	69		}
	70	    public void setStudyLevel(String studyLevel) {
	71	        this.studyLevel = studyLevel;
	72	    }
	73	    public void setCourseInterested(String courseInterested) {
	74	        this.courseInterested = courseInterested;
	75	    }
	76
	77
	78	    public static Student prepareEntity(StudentDTO dto) {
	79	        Student std = new Student();
	80	        std.setStudentId(dto.getStudentId());
	81	        std.setStudentName(dto.getStudentName());
	82	        std.setEmilId(dto.getEmilId());
	83	        std.setInterestedCountry(dto.getInterestedCountry());
	84	        std.setIntakeYear(dto.getIntakeYear());
	85	        std.setStudyLevel(dto.getStudyLevel());
	86	        std.setCourseInterested(dto.getCourseInterested());
	87	        return std;
	88
	89	    }
	90
	91	    public static StudentDTO prepareDTO(Student std) {
	92	        StudentDTO dto = new StudentDTO();
	93	        dto.setStudentId(std.getStudentId());
	94	        dto.setStudentName(std.getStudentName());
	95	        dto.setEmilId(std.getEmilId());
	96	        dto.setInterestedCountry(std.getInterestedCountry());
	97	        dto.setIntakeYear(std.getIntakeYear());
	98	        dto.setStudyLevel(std.getStudyLevel());
	99	        dto.setCourseInterested(std.getCourseInterested());
	100	        return dto;
	101	    }
	102
	103
	104
	105	}
	106
```
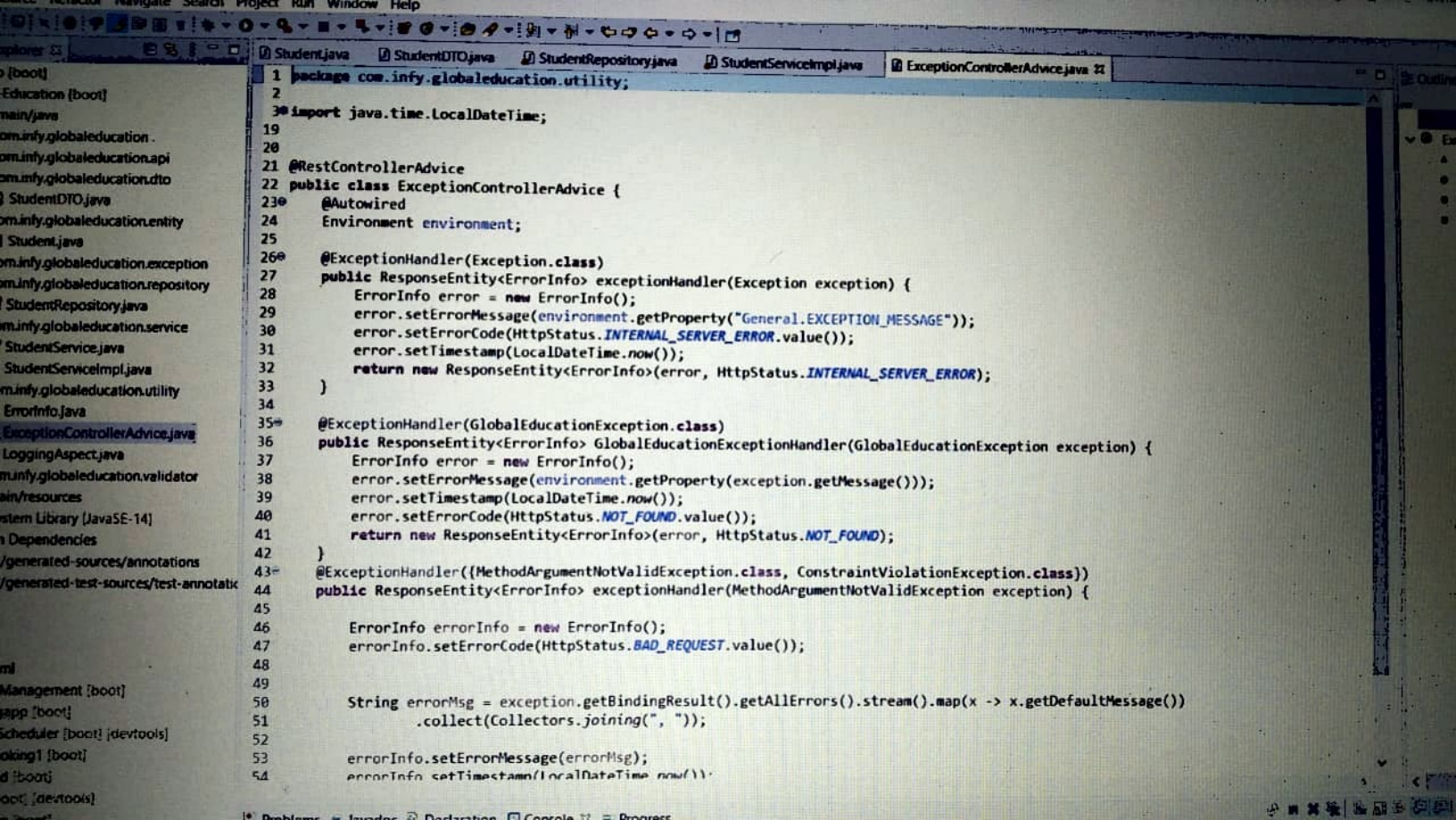
```java
2
3 import java.util.ArrayList;

17
18 @Service(value = "studentService")
19 @Transactional
20 public class StudentServiceImpl implements StudentService {
21
22     @Autowired
23     private StudentRepository studentrepo;
24
25     @Override
26     public StudentDTO registerStudent(StudentDTO studentDTO) throws GlobalEducationException {
27         StudentValidator.validateStudent(studentDTO);
28         Optional<Student> optional = studentrepo.findByEmilId(studentDTO.getEmilId());
29
30         if(optional.isPresent()) {
31             throw new GlobalEducationException("StudentService.STUDENT_ALREADY_EXISTS");
32         }
33         Student std = StudentDTO.prepareEntity(studentDTO);
34         std = studentrepo.save(std);
35
36         studentDTO.setStudentId(std.getStudentId());
37         return studentDTO;
38     }
39     @Override
40     public List<StudentDTO> findStudentByInterestedCountryAndIntakeYear(String country, [Return type of 'findStudentByInterestedCountryAndIntakeYear()']
41         throws GlobalEducationException {
42
43         List<Student> std = studentrepo.findStudentByInterestedCountryAndIntakeYear(country, intakeYear);
44         if (std.isEmpty() || std.size() == 0) {
45             throw new GlobalEducationException("StudentService.NO_STUDENTS_FOUND");
46         }
47         List<StudentDTO> dtos = new ArrayList<>();
48         std.stream().forEach(x -> dtos.add(new StudentDTO().prepareDTO(x)));
49         dtos.sort((x1,x2) -> x1.getStudentName().compareTo(x2.getStudentName()));
50
51         return dtos;
52     }
53 }
```

```java
package com.infy.globaleducation.utility;

import java.time.LocalDateTime;

@RestControllerAdvice
public class ExceptionControllerAdvice {
    @Autowired
    Environment environment;

    @ExceptionHandler(Exception.class)
    public ResponseEntity<ErrorInfo> exceptionHandler(Exception exception) {
        ErrorInfo error = new ErrorInfo();
        error.setErrorMessage(environment.getProperty("General.EXCEPTION_MESSAGE"));
        error.setErrorCode(HttpStatus.INTERNAL_SERVER_ERROR.value());
        error.setTimestamp(LocalDateTime.now());
        return new ResponseEntity<ErrorInfo>(error, HttpStatus.INTERNAL_SERVER_ERROR);
    }

    @ExceptionHandler(GlobalEducationException.class)
    public ResponseEntity<ErrorInfo> GlobalEducationExceptionHandler(GlobalEducationException exception) {
        ErrorInfo error = new ErrorInfo();
        error.setErrorMessage(environment.getProperty(exception.getMessage()));
        error.setTimestamp(LocalDateTime.now());
        error.setErrorCode(HttpStatus.NOT_FOUND.value());
        return new ResponseEntity<ErrorInfo>(error, HttpStatus.NOT_FOUND);
    }
    @ExceptionHandler({MethodArgumentNotValidException.class, ConstraintViolationException.class})
    public ResponseEntity<ErrorInfo> exceptionHandler(MethodArgumentNotValidException exception) {

        ErrorInfo errorInfo = new ErrorInfo();
        errorInfo.setErrorCode(HttpStatus.BAD_REQUEST.value());


        String errorMsg = exception.getBindingResult().getAllErrors().stream().map(x -> x.getDefaultMessage())
                .collect(Collectors.joining(", "));

        errorInfo.setErrorMessage(errorMsg);
        errorInfo.setTimestamp(LocalDateTime.now());
```

Student.java StudentDTO.java StudentRepository.java StudentServiceImpl.java ExceptionControllerAdvice.java

```java
21  @RestControllerAdvice
22  public class ExceptionControllerAdvice {
23      @Autowired
24      Environment environment;
25
26      @ExceptionHandler(Exception.class)
27      public ResponseEntity<ErrorInfo> exceptionHandler(Exception exception) {
28          ErrorInfo error = new ErrorInfo();
29          error.setErrorMessage(environment.getProperty("General.EXCEPTION_MESSAGE"));
30          error.setErrorCode(HttpStatus.INTERNAL_SERVER_ERROR.value());
31          error.setTimestamp(LocalDateTime.now());
32          return new ResponseEntity<ErrorInfo>(error, HttpStatus.INTERNAL_SERVER_ERROR);
33      }
34
35      @ExceptionHandler(GlobalEducationException.class)
36      public ResponseEntity<ErrorInfo> GlobalEducationExceptionHandler(GlobalEducationException exception) {
37          ErrorInfo error = new ErrorInfo();
38          error.setErrorMessage(environment.getProperty(exception.getMessage()));
39          error.setTimestamp(LocalDateTime.now());
40          error.setErrorCode(HttpStatus.NOT_FOUND.value());
41          return new ResponseEntity<ErrorInfo>(error, HttpStatus.NOT_FOUND);
42      }
43      @ExceptionHandler({MethodArgumentNotValidException.class, ConstraintViolationException.class})
44      public ResponseEntity<ErrorInfo> exceptionHandler(MethodArgumentNotValidException exception) {
45
46          ErrorInfo errorInfo = new ErrorInfo();
47          errorInfo.setErrorCode(HttpStatus.BAD_REQUEST.value());
48
49
50          String errorMsg = exception.getBindingResult().getAllErrors().stream().map(x -> x.getDefaultMessage())
51                  .collect(Collectors.joining(", "));
52
53          errorInfo.setErrorMessage(errorMsg);
54          errorInfo.setTimestamp(LocalDateTime.now());
55          return new ResponseEntity<>(errorInfo, HttpStatus.BAD_REQUEST);
56      }
57
58  }
59
```

```java
1  package com.infy.globaleducation.validator;
2
3  import java.time.LocalDate;

9
10 public class StudentValidator {
11
12     public static void validateStudent(StudentDTO studentDTO) throws GlobalEducationException {
13         if (! isValidIntakeYear(studentDTO.getIntakeYear())) {
14             throw new GlobalEducationException("StudentValidator.INVALID_INTAKE_YEAR");
15         }
16
17     }
18
19     public static Boolean isValidIntakeYear(Integer intakeYear) {
20         //int currentYear = new Date().getYear();
21
22
23         if (intakeYear < LocalDate.now().getYear()) {
24             return false;
25         }
26         else {
27             return true;
28         }
29     }
30 }
31
```

```java
 1  package com.infy.globaleducation.api;
 2  import java.util.List;
21
22  @RestController
23  @RequestMapping(value="/global-edu")
24  @Validated
25  public class StudentAPI  {
26
27      @Autowired
28      private StudentService studentservice;
29
30          @PostMapping(value="/student")
31      public ResponseEntity<StudentDTO> regiserStudent(@Valid @RequestBody StudentDTO studentDTO)
32              throws GlobalEducationException{
33          StudentDTO studentdto=studentservice.registerStudent(studentDTO);
34          return new ResponseEntity<>(studentdto,HttpStatus.CREATED);
35
36      }
37
38      @GetMapping(value="/student/{country}/{intakeYear}")
39      public ResponseEntity<List<StudentDTO>> findStudentByInterestedCountryAndIntakeYear(@PathVariable String country ,
40              @PathVariable @Pattern(regexp = "[0-9]{4}", message="{Student.intakeyear.Invalid}") Integer intakeYear)
41                  throws GlobalEducationException{
42          List<StudentDTO> studentdto1 =studentservice.findStudentByInterestedCountryAndIntakeYear(country, intakeYear);
43          return new ResponseEntity<>(studentdto1,HttpStatus.OK) ;
44
45      }
46
47  }
48
```