http://localhost:3030/api/accounts/signup

For SignUp -> Post method -> Body -> x-www-form-urlencoded

2-

```
{
	email: negin@gmail.com
	password: abc123
	name: negin mortazavi
}
```

3-

```
{
	email: alex@gmail.com
	password: alex123
	name: Alex
}
```

4-

```
{
	email: emy@gmail.com
	password: emy123
	name: Emy
}
```

5-

```
{
	email: christine@yahoo.com
	password: christine_1
	name: Christine
}
```

6-

```
{
	email: web@development.com
	password: cs5610
	name: Web Project
}
```

7-

```
{
	email: john@husky.neu.edu
	password: 9876
	name: John Smith
}
```

8-

```
{
	email: phil@outlook.com
	password: !ab492
	name: Philip Gust
}
```

9-

```
{
	email: course@outlook.com
	password: course
	name: Negin Alex Christine
}
```

10-

```
{
	email: peter@outlook.com
	password: 1p2e3t4e5r
	name: Peter Johnson
}
```

---

http://localhost:3030/api/accounts/login

For Login -> Post method -> Body -> x-www-form-urlencoded

1-
email: negin@gmail.com

password: abc123

since it is in database, you will receive:

```
{
   "success": true,
   "mesage": "You got your token",
   "token":
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyIjp7ImFkZHJlc3MiOnsiY2l0eSI6
ImF3c29tZSIsImNvdW50cnkiOiJVU0EiLCJzdGF0ZSI6IkNBIn0sImNyZWF0ZWQiOiI
yMDE4LTA0LTEzVDA0OjUxOjQyLjU1OFoiLCJfaWQiOiI1YWQwMzc1ZWQ3YWUxMz
E1NzZmZWQ5NTciLCJuYW1lIjoibmVnaW4gbW9ydGF6YXZpIiwiZW1haWwiOiJuZW
dpbkBnbWFpbC5jb20iLCJwYXNzd29yZCI6IiQyYSQxMCRMVmRGQXdKdWN3dHR
Ka2J0SVVCLnYuLkFqR1hFVnhEQnExQ1IyMWN2aXdaWDl1U280QWJULiIsInBpY3R1
cmUiOiJodHRwczovL2dyYXZhdGFyLmNvbS9hdmF0YXIvNGFkY2NhNDliM2IxZTVh
MDhhYzIwMmY1ZDVhOWU2ODg_czIwMCZkPXJldHJvIiwiX192IjowfSwiaWF0IjoxNT
IzNjU3MDcxLCJleHAiOjE1MjQyNjE4NzF9.AlYSW125VMfrLp_QrJvrKVpai_-
VDdwntM8jVUF77PQ"
}
```

2-
email: negin1@gmail.com
password: abc123

since email is wrong you will receive:

```
{
   "success": false,
   "message": "Login failed, Student not found"
}
```

3-
email: negin1@gmail.com
password: abc1234

since password is wrong, you will receive:

```
{
   "success": false,
   "message": "Login failed. You entered wrong password"
}
```

---

http://localhost:3030/api/accounts/profile

for retreiving the profile:

GET method -> Headers ->
Authorization:     provide the token from one of the users
Body -> x-www-form-urlencoded

email: negin@gmail.com
password: abc123

then it will give you the following:

```
{
    "success": true,
    "user": {
        "address": {
            "city": "awsome",
            "country": "USA",
            "state": "CA"
        },
        "created": "2018-04-13T04:51:42.558Z",
        "_id": "5ad0375ed7ae131576fed957",
        "name": "negin mortazavi",
        "email": "negin@gmail.com",
        "password":
"$2a$10$LVdFAwJucwttJkbtIUB.v..AjGXEVxDBq1CR21cviwZX9uSo4AbT.",
        "picture": "https://gravatar.com/avatar/
4adcca49b3b1e5a08ac202f5d5a9e688?s200&d=retro",
        "__v": 0
    },
    "message": "Successful"
}
```

if you do not provide the token:

```
{
    "success": false,
    "message": "No token provided"
}
```

and if the token you provided is wrong or the user with that token is not in database:

```
{
    "success": false,
    "message": "Failed to authenticate token"
}
```

  –  - - - - - - - - - - - -

for editing the profile:
Post method -> Headers ->
Authorization:     provide the token from one of the users
Body -> x-www-form-urlencoded

```
{
        email: abc@gmail.com
        password: abc123
        name: mr ABC
}
```

then you will get:

```
{
    "success": true,
    "message": "You successfully edited your profile"
}
```

if you do not provide token:

```
{
    "success": false,
    "message": "No token provided"
}
```

---

For retreiving the semesters
http://localhost:3030/api/semesters

GET Method -> no need for token

```
{
    "success": true,
    "message": "Success",
    "semesters": [
        {
            "created": "2018-04-14T20:39:22.422Z",
            "_id": "5ad266fa13baf31f6bbfaee9",
    "name": "Spring 2018",
    "__v": 0
  },
  {
    "created": "2018-04-14T20:41:00.225Z",
    "_id": "5ad2675c13baf31f6bbfaeeb",
    "name": "Fall 2017",
    "__v": 0
  },
  {
    "created": "2018-04-14T20:41:06.619Z",
    "_id": "5ad2676213baf31f6bbfaeec",
    "name": "Summer 2017",
    "__v": 0
  },
  {
    "created": "2018-04-14T20:41:16.542Z",
    "_id": "5ad2676c13baf31f6bbfaeed",
    "name": "Spring 2017",
    "__v": 0
  },
  {
    "created": "2018-04-14T20:41:21.883Z",
    "_id": "5ad2677113baf31f6bbfaeee",
    "name": "Fall 2016",
    "__v": 0
  },
  {
    "created": "2018-04-14T20:41:30.817Z",
    "_id": "5ad2677a13baf31f6bbfaeef",
    "name": "Summer 2016",
    "__v": 0
  },
  {
    "created": "2018-04-14T20:42:16.664Z",
    "_id": "5ad267a813baf31f6bbfaef0",
    "name": "Spring 2016",
    "__v": 0
  },
  {
    "created": "2018-04-14T20:42:23.146Z",
    "_id": "5ad267af13baf31f6bbfaef1",
    "name": "Fall 2015",
            "__v": 0
        },
        {
            "created": "2018-04-14T20:42:32.078Z",
            "_id": "5ad267b813baf31f6bbfaef2",
            "name": "Summer 2015",
            "__v": 0
        },
        {
            "created": "2018-04-14T20:43:07.741Z",
            "_id": "5ad267db13baf31f6bbfaef3",
            "name": "Spring 2015",
            "__v": 0
        }
    ]
}
```

POST Method -> Body -> x-www-form-urlencoded

name      Fall 2016

---

http://localhost:3030/api/courses
For retreiving the courses, to get all the courses
GET Method -> no need for token

```
{
    "success": true,
    "message": "semester",
    "courses": [
        {
            "reviews": [],
            "created": "2018-04-16T02:20:58.091Z",
            "_id": "5ad4088aa43cb02510b9d0ec",
            "name": "Discrete and Data Structures",
            "instructor": "Philip James Gust ",
            "description": "Introduces the mathematical structures and methods that form the foundation of computer science. Studies structures such as sets, tuples, sequences, lists, trees, and graphs. Discusses functions, relations, ordering, and equivalence relations. Examines inductive and recursive definitions of structures and functions. Covers principles of proof such as truth tables, inductive proof, and basic logic and the counting techniques and arguments needed to estimate the size of sets, the growth of functions, and the space-time complexity of algorithms.
```

```
Also, discusses data structures such as arrays, stacks, queues, lists, and the algorithms that manipulate them. ",
            "number": "CS5002",
            "credit": 4,
            "semester": {
                "created": "2018-04-14T20:41:00.225Z",
                "_id": "5ad2675c13baf31f6bbfaeeb",
                "name": "Fall 2017",
                "__v": 0
            },
            "__v": 1,
            "review": [
                "5ad4da6d294de029742316ae"
            ],
            "id": "5ad4088aa43cb02510b9d0ec"
        },
        {
            "reviews": [],
            "created": "2018-04-16T02:22:03.130Z",
            "_id": "5ad408cba43cb02510b9d0ed",
            "name": "Intensive Foundations of Computer Science",
            "instructor": "Philip James Gust ",
            "description": "Introduces the fundamental ideas of computing and programming principles. Discusses a systematic approach to word problems, including analytic reading, synthesis, goal setting, planning, plan execution, and testing. Presents several models of computing, beginning with functional program design. The latter part of the course consists of two parts: a task organization (ranging from the description of data to the creation of a test suite) and a data-oriented approach to the organization of programs (ranging from atomic data to self-referential data definitions and functions as data). Offers students an opportunity to practice pair programming and public code review techniques, as found in industry today. No prior programming experience is assumed; therefore, suitable for students with little or no computer science background. ",
            "number": "CS5001",
            "credit": 4,
            "semester": {
                "created": "2018-04-14T20:41:00.225Z",
                "_id": "5ad2675c13baf31f6bbfaeeb",
                "name": "Fall 2017",
                "__v": 0
            },
            "__v": 0,
            "id": "5ad408cba43cb02510b9d0ed"
```

```
    },
    {
        "reviews": [],
        "created": "2018-04-16T02:23:28.523Z",
        "_id": "5ad40920a43cb02510b9d0ee",
        "name": "Computer Science and Its Applications",
        "instructor": "Karl J Lieberherr",
        "description": "Introduces students to the field of computer science and
the patterns of thinking that enable them to become intelligent users of software
tools in a problem-solving setting. Examines several important software
applications so that students may develop the skills necessary to use computers
effectively in their own disciplines. ",
        "number": "CS1100",
        "credit": 4,
        "semester": {
            "created": "2018-04-14T20:41:00.225Z",
            "_id": "5ad2675c13baf31f6bbfaeeb",
            "name": "Fall 2017",
            "__v": 0
        },
        "__v": 0,
        "id": "5ad40920a43cb02510b9d0ee"
    },
    {
        "reviews": [],
        "created": "2018-04-16T02:25:17.297Z",
        "_id": "5ad4098da43cb02510b9d0ef",
        "name": "Logic and Computation",
        "instructor": "David William Sprague",
        "description": "Introduces formal logic and its connections to computer and
information science. Offers an opportunity to learn to translate statements about
the behavior of computer programs into logical claims and to gain the ability to
prove such assertions both by hand and using automated tools. Considers
approaches to proving termination, correctness, and safety for programs.
Discusses notations used in logic, propositional and first order logic, logical
inference, mathematical induction, and structural induction. Introduces the use of
logic for modeling the range of artifacts and phenomena that arise in computer
and information science. ",
        "number": "CS2800",
        "credit": 4,
        "semester": {
            "created": "2018-04-14T20:41:00.225Z",
            "_id": "5ad2675c13baf31f6bbfaeeb",
                "name": "Fall 2017",
                "__v": 0
            },
            "__v": 0,
            "id": "5ad4098da43cb02510b9d0ef"
        },
        {
            "reviews": [],
            "created": "2018-04-16T02:26:06.589Z",
            "_id": "5ad409bea43cb02510b9d0f0",
            "name": "Lab for CS 2800",
            "instructor": "David William Sprague",
            "description": "Introduces formal logic and its connections to computer and
information science. Offers an opportunity to learn to translate statements about
the behavior of computer programs into logical claims and to gain the ability to
prove such assertions both by hand and using automated tools. Considers
approaches to proving termination, correctness, and safety for programs.
Discusses notations used in logic, propositional and first order logic, logical
inference, mathematical induction, and structural induction. Introduces the use of
logic for modeling the range of artifacts and phenomena that arise in computer
and information science. ",
            "number": "CS281",
            "credit": 1,
            "semester": {
                "created": "2018-04-14T20:41:00.225Z",
                "_id": "5ad2675c13baf31f6bbfaeeb",
                "name": "Fall 2017",
                "__v": 0
            },
            "__v": 0,
            "id": "5ad409bea43cb02510b9d0f0"
        },
        {
            "reviews": [],
            "created": "2018-04-16T02:26:58.139Z",
            "_id": "5ad409f2a43cb02510b9d0f1",
            "name": "Database Design",
            "instructor": "Ghita Amor-Tijani",
            "description": "Studies the design of a database for use in a relational
database management system. The entity-relationship model and normalization
are used in problems. Relational algebra and then the SQL (structured query
language) are presented. Advanced topics include triggers, stored procedures,
indexing, elementary query optimization, and fundamentals of concurrency and
```

```
recovery. Students implement a database schema and short application programs
on one or more commercial relational database management systems. ",
      "number": "CS3200",
      "credit": 4,
      "semester": {
        "created": "2018-04-14T20:41:00.225Z",
        "_id": "5ad2675c13baf31f6bbfaeeb",
        "name": "Fall 2017",
        "__v": 0
      },
      "__v": 0,
      "id": "5ad409f2a43cb02510b9d0f1"
    },
    {
      "reviews": [],
      "created": "2018-04-16T06:44:33.662Z",
      "_id": "5ad44651270c1d2702d86cb9",
      "__v": 0,
      "id": "5ad44651270c1d2702d86cb9"
    },
    {
      "reviews": [],
      "created": "2018-04-16T02:27:36.983Z",
      "_id": "5ad40a18a43cb02510b9d0f2",
      "name": "Object-Oriented Design",
      "instructor": "Amit Prakash Shesh",
      "description": "Presents a comparative approach to object-oriented
programming and design. Discusses the concepts of object, class, meta-class,
message, method, inheritance, and genericity. Reviews forms of polymorphism in
object-oriented languages. Contrasts the use of inheritance and composition as
dual techniques for software reuse: forwarding vs. delegation and subclassing vs.
subtyping. Fosters a deeper understanding of the principles of object-oriented
programming and design including software components, object-oriented design
patterns, and the use of graphical design notations such as UML (unified modeling
language). Basic concepts in object-oriented design are illustrated with case
studies in application frameworks and by writing programs in one or more object-
oriented languages. ",
      "number": "CS3500",
      "credit": 4,
      "semester": {
        "created": "2018-04-14T20:41:00.225Z",
        "_id": "5ad2675c13baf31f6bbfaeeb",
        "name": "Fall 2017",
        "__v": 0
      },
      "__v": 0,
      "id": "5ad40a18a43cb02510b9d0f2"
    },
    {
      "reviews": [],
      "created": "2018-04-16T02:28:34.835Z",
      "_id": "5ad40a52a43cb02510b9d0f3",
      "name": "Programming in C++",
      "instructor": "Seth Everett Cooper",
      "description": "Examines how to program in C++ in a robust and safe
manner. Reviews basics, including scoping, typing, and primitive data structures.
Discusses data types (primitive, array, structure, class, string); addressing/
parameter mechanisms (value, pointer, reference); stacks; queues; linked lists;
binary trees; hash tables; and the design of classes and class inheritance,
emphasizing single inheritance. Considers the instantiation of objects, the trade-
offs of stack vs. heap allocation, and the design of constructors and destructors.
Emphasizes the need for a strategy for dynamic memory management. Addresses
function and operator overloading; templates, the Standard Template Library
(STL), and the STL components (containers, generic algorithms, iterators,
adaptors, allocators, function objects); streams; exception handling; and system
calls for processes and threads. ",
      "number": "CS3520",
      "credit": 4,
      "semester": {
        "created": "2018-04-14T20:41:00.225Z",
        "_id": "5ad2675c13baf31f6bbfaeeb",
        "name": "Fall 2017",
        "__v": 0
      },
      "__v": 0,
      "id": "5ad40a52a43cb02510b9d0f3"
    },
    {
      "reviews": [],
      "created": "2018-04-16T02:32:00.052Z",
      "_id": "5ad40b20a43cb02510b9d0f5",
      "name": "Computer Science/Information Science Overview 1",
      "instructor": "Megan Michelle Giordano ",
      "description": "Introduces students to the College of Computer and
Information Science (CCIS) and begins their preparation for careers in the
computing and information fields. Offers students an opportunity to learn how to
```

```
thrive at Northeastern and within CCIS by developing academic, professional, and
interpersonal skills. Covers the variety of careers available in the high-technology
professions. Students work in groups to create and deliver presentations on
careers in the field. Intended for freshmen.",
      "number": "CS1200",
      "credit": 1,
      "semester": {
        "created": "2018-04-14T20:39:22.422Z",
        "_id": "5ad266fa13baf31f6bbfaee9",
        "name": "Spring 2018",
        "__v": 0
      },
      "__v": 0,
      "id": "5ad40b20a43cb02510b9d0f5"
    }
  ],
  "totalCourses": 24,
  "pages": 3
}
```

for posting the courses:
POST method:
http://localhost:3030/api/course
POST Method -> Body -> x-www-form-urlencoded

name
instructor
description
number
semesterId:   find id of one semester
credit

To get just one course based on the id of that course:   GET Method
http://localhost:3030/api/course/5ad4088aa43cb02510b9d0ec

To retreive the list of courses for specific semester:
http://localhost:3030/api/semesters/5ad267b813baf31f6bbfaef2
no need for token

---

To post the sales of student for any course note or book:

http://localhost:3030/api/post/sale

POST method:
Authorization: token related to login information
title:
description
rating
price
sale_picture
courseId
edition
author

To retrieve the information of sales for specific owner(student):

http://localhost:3030/api/post/sale
GET method:

```
{
    "success": true,
    "message": "Sales",
    "sales": [
        {
            "created": "2018-04-16T06:50:57.256Z",
            "_id": "5ad447d1fcc624271189bb54",
            "owner": {
                "address": {
                    "city": "awsome",
                    "country": "USA",
                    "state": "CA"
                },
                "created": "2018-04-13T04:51:42.558Z",
                "_id": "5ad0375ed7ae131576fed957",
                "name": "negin mortazavi",
                "email": "negin@gmail.com",
                "password":
"$2a$10$LVdFAwJucwttJkbtIUB.v..AjGXEVxDBq1CR21cviwZX9uSo4AbT.",
                "picture": "https://gravatar.com/avatar/
4adcca49b3b1e5a08ac202f5d5a9e688?s200&d=retro",
                "__v": 0
            },
            "course": {
```

```
        "created": "2018-04-16T02:20:58.091Z",
        "_id": "5ad4088aa43cb02510b9d0ec",
        "name": "Discrete and Data Structures",
        "instructor": "Philip James Gust ",
        "description": "Introduces the mathematical structures and methods that form the foundation of computer science. Studies structures such as sets, tuples, sequences, lists, trees, and graphs. Discusses functions, relations, ordering, and equivalence relations. Examines inductive and recursive definitions of structures and functions. Covers principles of proof such as truth tables, inductive proof, and basic logic and the counting techniques and arguments needed to estimate the size of sets, the growth of functions, and the space-time complexity of algorithms. Also, discusses data structures such as arrays, stacks, queues, lists, and the algorithms that manipulate them. ",
        "number": "CS5002",
        "credit": 4,
        "semester": "5ad2675c13baf31f6bbfaeeb",
        "__v": 0
      },
      "title": "Data Structures and Algorithms in Java",
      "author": "Michael T. Goodrich, Roberto Tamassia, Michael H. Goldwasser",
      "price": 168,
      "edition": 6,
      "rating": "very good",
      "description": "The design and analysis of efficient data structures has long been recognized as a key component of the Computer Science curriculum. Goodrich, Tomassia and Goldwasser's approach to this classic topic is based on the object-oriented paradigm as the framework of choice for the design of data structures. For each ADT presented in the text, the authors provide an associated Java interface. Concrete data structures realizing the ADTs are provided as Java classes implementing the interfaces. The Java code implementing fundamental data structures in this book is organized in a single Java package, net.datastructures. This package forms a coherent library of data structures and algorithms in Java specifically designed for educational purposes in a way that is complimentary with the Java Collections Framework.",
      "image": "https://webprojectcs5610.s3.amazonaws.com/1523861456664",
      "__v": 0
    },
    {
      "created": "2018-04-16T06:56:34.277Z",
      "_id": "5ad44922fcc624271189bb55",
      "owner": {
        "address": {
          "city": "awsome",
                "country": "USA",
                "state": "CA"
            },
            "created": "2018-04-13T04:51:42.558Z",
            "_id": "5ad0375ed7ae131576fed957",
            "name": "negin mortazavi",
            "email": "negin@gmail.com",
            "password":
"$2a$10$LVdFAwJucwttJkbtIUB.v..AjGXEVxDBq1CR21cviwZX9uSo4AbT.",
            "picture": "https://gravatar.com/avatar/
4adcca49b3b1e5a08ac202f5d5a9e688?s200&d=retro",
            "__v": 0
        },
        "course": {
            "created": "2018-04-16T02:20:58.091Z",
            "_id": "5ad4088aa43cb02510b9d0ec",
            "name": "Discrete and Data Structures",
            "instructor": "Philip James Gust ",
            "description": "Introduces the mathematical structures and methods that
form the foundation of computer science. Studies structures such as sets, tuples,
sequences, lists, trees, and graphs. Discusses functions, relations, ordering, and
equivalence relations. Examines inductive and recursive definitions of structures
and functions. Covers principles of proof such as truth tables, inductive proof, and
basic logic and the counting techniques and arguments needed to estimate the
size of sets, the growth of functions, and the space-time complexity of algorithms.
Also, discusses data structures such as arrays, stacks, queues, lists, and the
algorithms that manipulate them. ",
            "number": "CS5002",
            "credit": 4,
            "semester": "5ad2675c13baf31f6bbfaeeb",
            "__v": 0
        },
        "title": "Data Structures and Algorithms in Java",
        "author": "Michael T. Goodrich, Roberto Tamassia, Michael H. Goldwasser",
        "price": 168,
        "edition": 6,
        "rating": "very good",
        "description": "The design and analysis of efficient data structures has long
been recognized as a key component of the Computer Science curriculum.
Goodrich, Tomassia and Goldwasser's approach to this classic topic is based on
the object-oriented paradigm as the framework of choice for the design of data
structures. For each ADT presented in the text, the authors provide an associated
Java interface. Concrete data structures realizing the ADTs are provided as Java
```

```
classes implementing the interfaces. The Java code implementing fundamental
data structures in this book is organized in a single Java package,
net.datastructures. This package forms a coherent library of data structures and
algorithms in Java specifically designed for educational purposes in a way that is
complimentary with the Java Collections Framework.",
            "image": "https://webprojectcs5610.s3.us-west-1.amazonaws.com/
1523861793981",
            "__v": 0
        }
    ]
}
```

To retrieve the information of sales for specific course:

http://localhost:3030/api/post/sales/id of course
GET method:

```
{
    "success": true,
    "message": "Sales",
    "sales": [
        {
            "created": "2018-04-16T06:50:57.256Z",
            "_id": "5ad447d1fcc624271189bb54",
            "owner": {
                "address": {
                    "city": "awsome",
                    "country": "USA",
                    "state": "CA"
                },
                "created": "2018-04-13T04:51:42.558Z",
                "_id": "5ad0375ed7ae131576fed957",
                "name": "negin mortazavi",
                "email": "negin@gmail.com",
                "password":
"$2a$10$LVdFAwJucwttJkbtIUB.v..AjGXEVxDBq1CR21cviwZX9uSo4AbT.",
                "picture": "https://gravatar.com/avatar/
4adcca49b3b1e5a08ac202f5d5a9e688?s200&d=retro",
                "__v": 0
            },
            "course": {
                "created": "2018-04-16T02:20:58.091Z",
                "_id": "5ad4088aa43cb02510b9d0ec",
                "name": "Discrete and Data Structures",
                "instructor": "Philip James Gust ",
                "description": "Introduces the mathematical structures and methods that form the foundation of computer science. Studies structures such as sets, tuples, sequences, lists, trees, and graphs. Discusses functions, relations, ordering, and equivalence relations. Examines inductive and recursive definitions of structures and functions. Covers principles of proof such as truth tables, inductive proof, and basic logic and the counting techniques and arguments needed to estimate the size of sets, the growth of functions, and the space-time complexity of algorithms. Also, discusses data structures such as arrays, stacks, queues, lists, and the algorithms that manipulate them. ",
                "number": "CS5002",
                "credit": 4,
                "semester": "5ad2675c13baf31f6bbfaeeb",
                "__v": 0
            },
            "title": "Data Structures and Algorithms in Java",
            "author": "Michael T. Goodrich, Roberto Tamassia, Michael H. Goldwasser",
            "price": 168,
            "edition": 6,
            "rating": "very good",
            "description": "The design and analysis of efficient data structures has long been recognized as a key component of the Computer Science curriculum. Goodrich, Tomassia and Goldwasser's approach to this classic topic is based on the object-oriented paradigm as the framework of choice for the design of data structures. For each ADT presented in the text, the authors provide an associated Java interface. Concrete data structures realizing the ADTs are provided as Java classes implementing the interfaces. The Java code implementing fundamental data structures in this book is organized in a single Java package, net.datastructures. This package forms a coherent library of data structures and algorithms in Java specifically designed for educational purposes in a way that is complimentary with the Java Collections Framework.",
            "image": "https://webprojectcs5610.s3.amazonaws.com/1523861456664",
            "__v": 0
        }
    ]
}
```

To retreive all sales (book, note) for all students and courses

http://localhost:3030/api/post/sales

___________________

To post review for specific courses:

http://localhost:3030/api/review
It is the POST method, it needs the authorization token.
title:
description
rating
courseId

_____________________

Algolia account is the platform for serach engine
:
create account in Algolia
create indices: **web_development_project_student_forum**
**in API keys you ave all information which you will need later on in your code**
**now we need to install the library call mongoose-algolia so we can start doing search**

**npm install mongoose-algolia --save**

**add algolia to our course schema**

**now we should create the search API based on the Algolia, Algolia is the real time framework.**

first we should install all algolia official dependencies
npm install algoliasearch --save