# Assignment 2C

Addstudentcomponent.html:

Front-end

```
<div>
 <div class="submit-form">
  <div *ngIf="!submitted">
   <div class="form-group">
    <label for="name">Name</label>
    <input
     type="text"
     class="form-control"
     id="name"
     required
     [(ngModel)]="student.name"
     name="name"
    />
   </div>

   <div class="form-group">
    <label for="address">Address</label>
    <input
     class="form-control"
     id="address"
     required
     [(ngModel)]="student.address"
     name="address"
    />
   </div>

   <div class="form-group">
    <label for="email">Email</label>
    <input
     type="email"
     class="form-control”
     id="email"
     required
     [(ngModel)]="student.email"
     name="email"
    />
   </div>

   <div class="form-group">
    <label for="department">Department</label>
    <input
     class="form-control"
     id="department"
     required
     [(ngModel)]="student.department"
     name="department"
    />
   </div>

   <button (click)="saveStudent()" class="btn btn-
success">Submit</button>
  </div>

  <div *ngIf="submitted">
   <h4>Student was submitted successfully!</h4>
   <button class="btn btn-success"
(click)="newStudent()">Add</button>
  </div>
 </div>
</div>
```

Studentdetailscomponent.html

```
<div *ngIf="viewMode; else editable">
 <div *ngIf="currentStudent.id">
  <h4>Student</h4>
  <div>
   <label><strong>Name:</strong></label> {{
currentStudent.name }}
  </div>
  <div>
   <label><strong>Address:</strong></label>
   {{ currentStudent.address }}
  </div>
  <div>
   <label><strong>Email:</strong></label>
   {{ currentStudent.email }}
  </div>
  <div>
   <label><strong>Department:</strong></label>
   {{ currentStudent.department }}
  </div>
  <a
   class="badge badge-warning"
   routerLink="/students/{{ currentStudent.id }}"
  >
   Edit
  </a>
 </div>

 <div *ngIf="!currentStudent">
  <br />
  <p>Please click on a Student...</p>
 </div>
</div>
```

```
<ng-template #editable>
 <div *ngIf="currentStudent.id" class="edit-form">
  <h4>Student</h4>
  <form>
   <div class="form-group">
    <label for="name">name</label>
    <input
     type="text"
     class="form-control"
     id="name"
     [(ngModel)]="currentStudent.name"
     name="name"
    />
   </div>
   <div class="form-group">
    <label for="address">address</label>
    <input
     type="text"
     class="form-control"
     id="address"
     [(ngModel)]="currentStudent.address"
     name="address"
    />
   </div>
   <div class="form-group">
    <label for="email">Email</label>
    <input
     type="text"
     class="form-control"
     id="email"
     [(ngModel)]="currentStudent.email"
     name="email"
    />
    </div>
    <div class="form-group">
      <label for="department">Department</label>
      <input
        type="text"
        class="form-control"
        id="department"
        [(ngModel)]="currentStudent.department"
        name="department"
      />
    </div>
  </form>

  <button class="badge badge-danger mr-2"
(click)="deleteStudent()">
    Delete
  </button>

  <button
    type="submit"
    class="badge badge-success mb-2"
    (click)="updateStudent()"
  >
    Update
  </button>
  <p>{{ message }}</p>
 </div>

 <div *ngIf="!currentStudent.id">
  <br />
  <p>Cannot access this Student...</p>
 </div>
</ng-template>
```

<u>Studentlistcomponent.html</u>

```
<div class="list row">
 <div class="col-md-6">
  <h4>Students List</h4>
  <ul class="list-group">
   <li
    class="list-group-item"
    *ngFor="let student of students; let i = index"
    [class.active]="i == currentIndex"
    (click)="setActiveStudent(student, i)"
   >
    {{ student.name }}
   </li>
  </ul>

  <button class="m-3 btn btn-sm btn-danger"
(click)="removeAllStudents()">
   Remove All
  </button>
 </div>
 <div class="col-md-6">
  <app-student-details
   [viewMode]="true"
   [currentStudent]="currentStudent"
  ></app-student-details>
 </div>
</div>
```

**<u>Back-end</u>**

<u>Server.js</u>

```
<div class="list row">
 <div class="col-md-6">
  <h4>Students List</h4>
  <ul class="list-group">
    <li
      class="list-group-item"
      *ngFor="let student of students; let i = index"
      [class.active]="i == currentIndex"
      (click)="setActiveStudent(student, i)"
    >
      {{ student.name }}
    </li>
  </ul>

  <button class="m-3 btn btn-sm btn-danger"
(click)="removeAllStudents()">
    Remove All
  </button>
 </div>
 <div class="col-md-6">
  <app-student-details
    [viewMode]="true"
    [currentStudent]="currentStudent"
  ></app-student-details>
 </div>
</div>
```

INDEX.JS

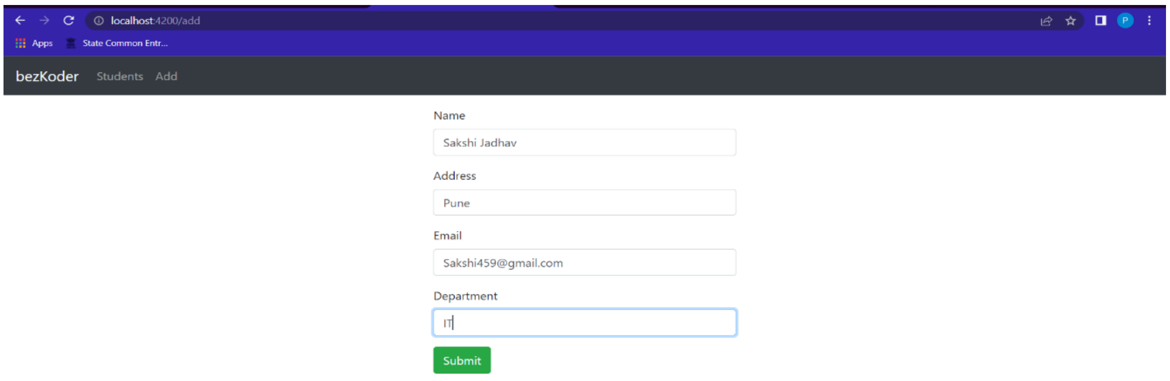
```
const dbConfig = require("../config/db.config.js");

const mongoose = require("mongoose");
mongoose.Promise = global.Promise;

const db = {};
db.mongoose = mongoose;
db.url = dbConfig.url;
db.students =
require("./student.model.js")(mongoose);

module.exports = db;
```

student module.js

```
module.exports = mongoose => {
  var schema = mongoose.Schema(
    {
      name: String,
      address: String,
      email: String,
      department: String
    },
    { timestamps: true }
  );

  schema.method("toJSON", function() {
    const { __v, _id, ...object } = this.toObject();
    object.id = _id;
    return object;
  });

  const Student = mongoose.model("student",
schema);
  return Student;
};
```

Student.controller.js

```
const db = require("../models");
const Student = db.students;

// Create and Save a new student
exports.create = (req, res) => {
  // Validate request
  if (!req.body.name) {
    res.status(400).send({ message: "Content can
not be empty!" });
    return;
  }

  // Create a student
  const student = new Student({
    name: req.body.name,
    address: req.body.address,
    email: req.body.email,
    department: req.body.department
  });

  // Save Student in the database
  student
    .save(student)
    .then(data => {
      res.send(data);
    })
    .catch(err => {
      res.status(500).send({
        message:
          err.message || "Some error occurred while
creating the Student."
      });
    });
};

// Retrieve all Student from the database.
exports.findAll = (req, res) => {
  const name = req.query.name;
  var condition = name ? { name: { $regex: new
RegExp(name), $options: "i" } } : {};

  Student.find(condition)
    .then(data => {
      res.send(data);
    })
    .catch(err => {
      res.status(500).send({
        message:
          err.message || "Some error occurred while
retrieving students."
      });
    });
};

// Find a single student with an id
exports.findOne = (req, res) => {
  const id = req.params.id;

  Student.findById(id)
    .then(data => {
      if (!data)
        res.status(404).send({ message: "Not found
student with id " + id });
      else res.send(data);
    })
    .catch(err => {
      res
        .status(500)
        .send({ message: "Error retrieving student
with id=" + id });
    });
};

// Update a student by the id in the request
exports.update = (req, res) => {
  if (!req.body) {
    return res.status(400).send({
      message: "Data to update can not be empty!"
    });
  }

  const id = req.params.id;

  Student.findByIdAndUpdate(id, req.body, {
useFindAndModify: false })
    .then(data => {
      if (!data) {
        res.status(404).send({
          message: `Cannot update student with
id=${id}. Maybe student was not found!`
        });
      } else res.send({ message: "student was
updated successfully." });
    })
    .catch(err => {
      res.status(500).send({
        message: "Error updating student with id=" +
id
      });
    });
};

// Delete a student with the specified id in the
request
exports.delete = (req, res) => {
  const id = req.params.id;

  Student.findByIdAndRemove(id, {
useFindAndModify: false })
    .then(data => {
      if (!data) {
        res.status(404).send({
          message: `Cannot delete student with
id=${id}. Maybe student was not found!`
        });
      } else {
        res.send({
          message: "student was deleted successfully!"
        });
      }
    })
    .catch(err => {
      res.status(500).send({
        message: "Could not delete Student with id="
+ id
      });
    });
};
```


localhost:4200/add

bezKoder Students Add

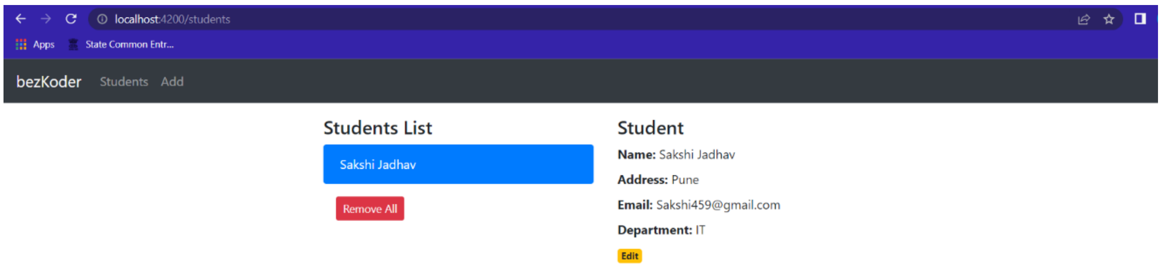
Name: Sakshi Jadhav

Address: Pune

Email: Sakshi459@gmail.com

Department: IT

Submit


localhost:4200/students

bezKoder Students Add

Students List

Sakshi Jadhav

Remove All

Student

**Name:** Sakshi Jadhav

**Address:** Pune

**Email:** Sakshi459@gmail.com

**Department:** IT

Edit