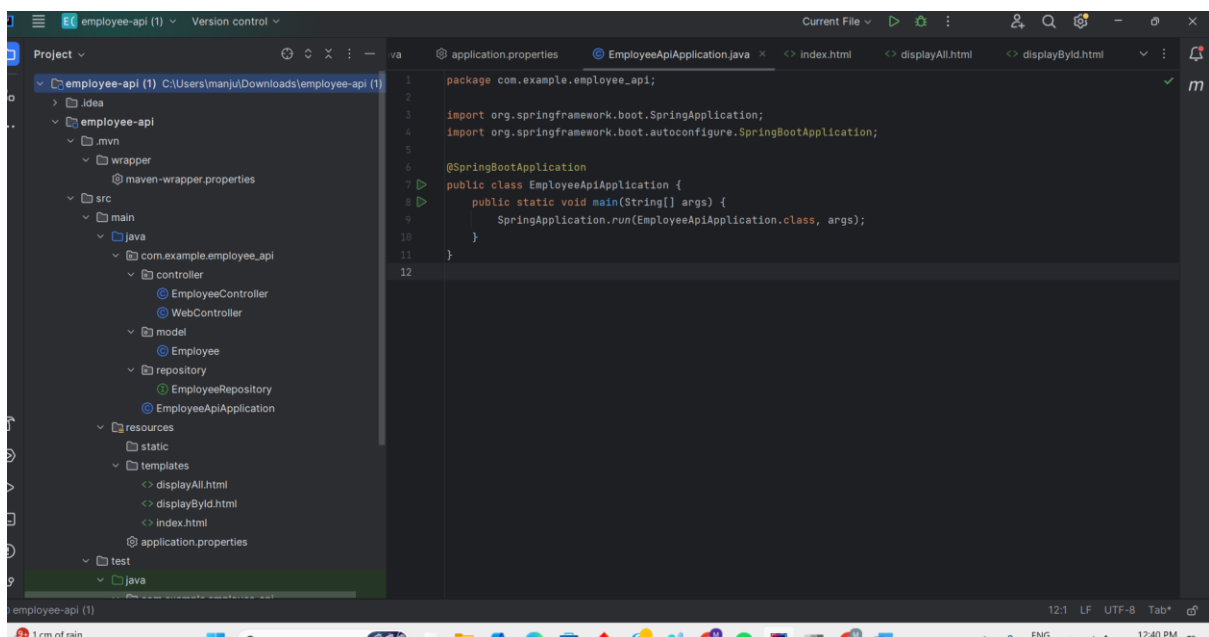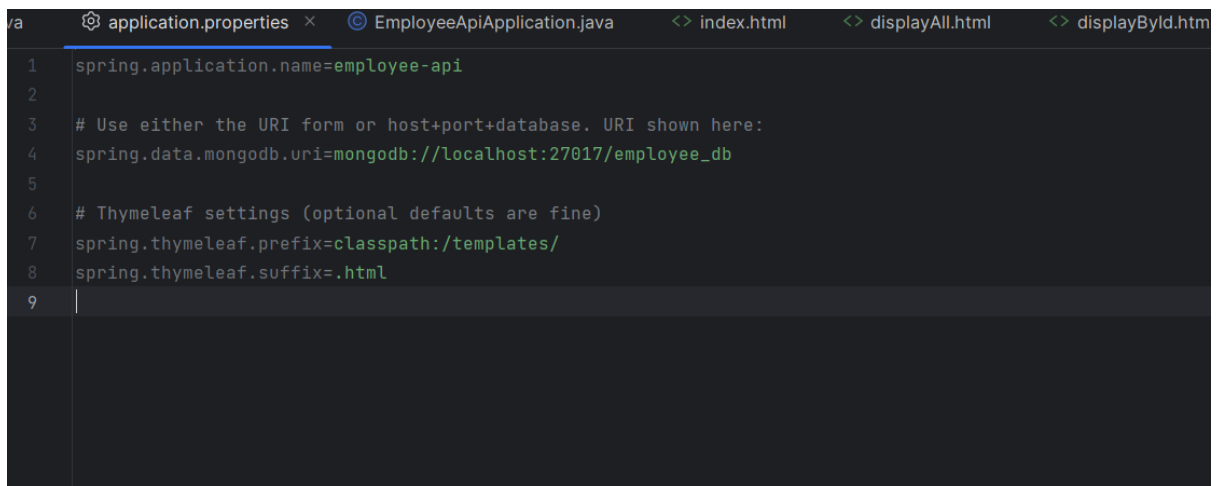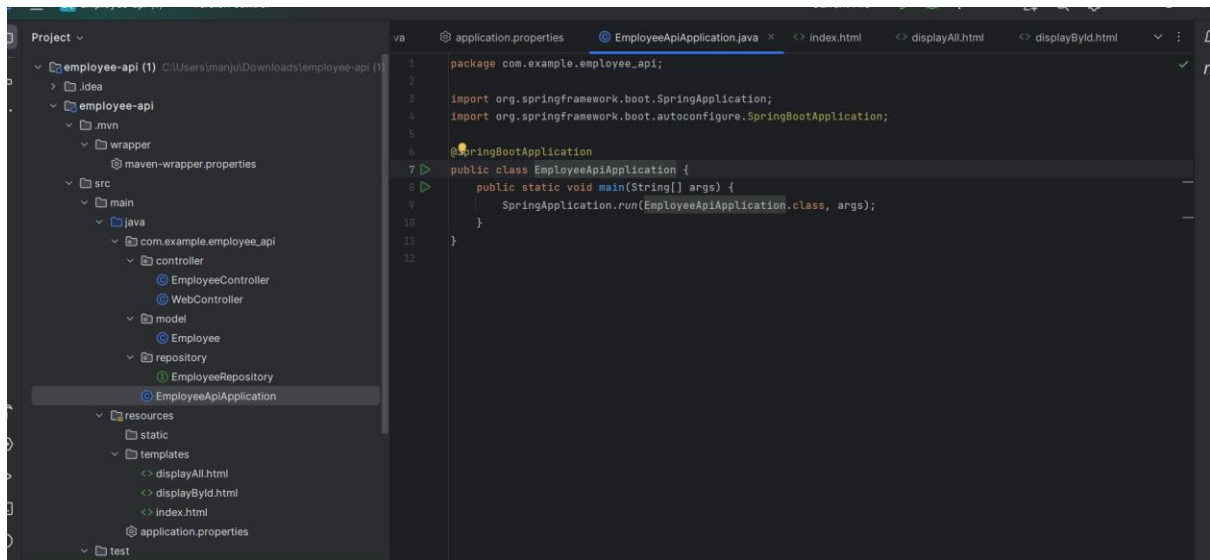# Spring Data Mongodb

1.Create a Restful API using spring mvc as per the requirements below:(should use mongodb)

1. The application should display a Form in the landing page (index.page) where the user can enter the 'Employee' details like 'Employee ID' , 'Employee Name' , 'Employee Email' and 'Location' and upon submission the data should be saved onto a database table

2. The application should also support a '/displayAll' path which shows the list of all the Employees.

3. The application should also support a '/display/<user id>' path where upon passing of userid as the parameter the details of only the particular Employee holding that Employee ID will be displayed.

4. Code all the above functionalities required as a RESTful URL.

```java
package com.example.employee_api;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class EmployeeApiApplication {
    public static void main(String[] args) {
        SpringApplication.run(EmployeeApiApplication.class, args);
    }
}
```

```properties
spring.application.name=employee-api

# Use either the URI form or host+port+database. URI shown here:
spring.data.mongodb.uri=mongodb://localhost:27017/employee_db

# Thymeleaf settings (optional defaults are fine)
spring.thymeleaf.prefix=classpath:/templates/
spring.thymeleaf.suffix=.html
```

```html
1   <!DOCTYPE html>
2   <html xmlns:th="http://www.thymeleaf.org">
3   <head>
4       <meta charset="utf-8"/>
5       <title>Add Employee</title>
6       <style>
7           body {
8               font-family: Arial, sans-serif;
9               margin: 40px;
10              background-color: #f9f9f9;
11          }
12          h2 {
13              text-align: center;
14              color: #333;
15          }
16          form {
17              max-width: 400px;
18              margin: auto;
19              padding: 20px;
20              background: #fff;
21              border-radius: 8px;
22              box-shadow: 0 2px 6px rgba(0,0,0,0.2);
23          }
24          label {
25              display: block;
26              margin: 10px 0 5px;
27              font-weight: bold;
28          }
29          input {
```

```html
52              margin-top: 20px;
53          }
54          .links a {
55              text-decoration: none;
56              color: #007bff;
57              font-size: 14px;
58          }
59      </style>
60  </head>
61  <body>
62  <h2>Add Employee</h2>
63
64  <form th:action="@{/employee/save}" th:object="${employee}" method="post">
65      <label>Employee ID:</label>
66      <input type="text" th:field="*{employeeId}" required/>
67
68      <label>Name:</label>
69      <input type="text" th:field="*{name}" required/>
70
71      <label>Email:</label>
72      <input type="email" th:field="*{email}" required/>
73
74      <label>Location:</label>
75      <input type="text" th:field="*{location}" required/>
76
77      <button type="submit">Submit</button>
78  </form>
79
80  <div class="links">
```

```html
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
    <meta charset="utf-8"/>
    <title>All Employees</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 40px;
            background: #f4f4f9;
        }
        h2 {
            text-align: center;
            color: #333;
        }
        form {
            text-align: center;
            margin-bottom: 20px;
        }
        input {
            padding: 6px;
            font-size: 14px;
            border: 1px solid #ccc;
            border-radius: 4px;
        }
        button {
            padding: 6px 12px;
            font-size: 14px;
            border: none;
```

```
31              background: #28a745;
32              color: white;
33              cursor: pointer;
34          }
35          button:hover {
36              background: #218838;
37          }
38          table {
39              width: 100%;
40              border-collapse: collapse;
41              margin-top: 15px;
42              background: white;
43          }
44          th, td {
45              border: 1px solid #ddd;
46              padding: 10px;
47              font-size: 14px;
48              text-align: center;
49          }
50          th {
51              background: #007bff;
52              color: white;
53          }
54          a {
55              text-decoration: none;
56              color: #007bff;
57          }
58          a:hover {
59              text-decoration: underline;
```

```
64          }
65      </style>
66  </head>
67  <body>
68  <h2>Employee List</h2>
69
70  <form th:action="@{/employee/display/search}" method="get">
71      <label>Search by Employee ID or Mongo ID:</label>
72      <input type="text" name="q" placeholder="Enter ID" required/>
73      <button type="submit">Search</button>
74  </form>
75
76  <table>
77      <thead>
78      <tr>
79          <th>Mongo _id</th>
80          <th>Employee ID</th>
81          <th>Name</th>
82          <th>Email</th>
83          <th>Location</th>
84          <th>Actions</th>
85      </tr>
86      </thead>
87      <tbody>
88      <tr th:each="emp : ${employees}">
89          <td th:text="${emp.id}"></td>
90          <td th:text="${emp.employeeId}"></td>
91          <td th:text="${emp.name}"></td>
92          <td th:text="${emp.email}"></td>
```

```html
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
    <meta charset="utf-8"/>
    <title>Employee Details</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 40px;
            background: #f4f4f9;
        }
        h2 {
            text-align: center;
            color: #333;
        }
        .card {
            max-width: 400px;
            margin: auto;
            padding: 20px;
            background: white;
            border-radius: 8px;
            box-shadow: 0 2px 6px rgba(0,0,0,0.2);
        }
        p {
            font-size: 15px;
            margin: 8px 0;
        }
        strong {
            color: #007bff;
```

```html
        .error {
            color: red;
            font-weight: bold;
            text-align: center;
        }
        .links {
            text-align: center;
            margin-top: 20px;
        }
        .links a {
            margin: 0 10px;
            text-decoration: none;
            color: #007bff;
        }
        .links a:hover {
            text-decoration: underline;
        }
    </style>
</head>
<body>
<h2>Employee Details</h2>

<div th:if="${employee != null}" class="card">
    <p><strong>Mongo _id:</strong> <span th:text="${employee.id}"></span></p>
    <p><strong>Employee ID:</strong> <span th:text="${employee.employeeId}"></span></p>
    <p><strong>Name:</strong> <span th:text="${employee.name}"></span></p>
    <p><strong>Email:</strong> <span th:text="${employee.email}"></span></p>
    <p><strong>Location:</strong> <span th:text="${employee.location}"></span></p>
</div>
```

```
48          </style>
49    </head>
50    <body>
51    <h2>Employee Details</h2>
52
53    <div th:if="${employee != null}" class="card">
54        <p><strong>Mongo _id:</strong> <span th:text="${employee.id}"></span></p>
55        <p><strong>Employee ID:</strong> <span th:text="${employee.employeeId}"></span></p>
56        <p><strong>Name:</strong> <span th:text="${employee.name}"></span></p>
57        <p><strong>Email:</strong> <span th:text="${employee.email}"></span></p>
58        <p><strong>Location:</strong> <span th:text="${employee.location}"></span></p>
59    </div>
60
61    <div th:if="${employee == null}" class="error">
62        ❌ No employee found with that identifier.
63    </div>
64
65    <div class="links">
66        <a th:href="@{/employee/displayAll}">Back to All</a> |
67        <a th:href="@{/employee/}">Add New</a>
68    </div>
69    </body>
70    </html>
71
```

```java
package com.example.employee_api.repository;

import com.example.employee_api.model.Employee;
import org.springframework.data.mongodb.repository.MongoRepository;

import java.util.Optional;

4 usages
public interface EmployeeRepository extends MongoRepository<Employee, String> {
    2 usages
    Optional<Employee> findByEmployeeId(String employeeId);
}
```

```java
package com.example.employee_api.controller;

import com.example.employee_api.model.Employee;
import com.example.employee_api.repository.EmployeeRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;

import java.util.List;
import java.util.Optional;

no usages
@RestController
@RequestMapping("/api/employee")
public class EmployeeController {

    5 usages
    @Autowired
    private EmployeeRepository repo;

    no usages
    @PostMapping("/add")
    public Employee addEmployee(@RequestBody Employee emp) {
        return repo.save(emp);
    }

    no usages
    @GetMapping("/all")
    public List<Employee> all() {
        return repo.findAll();
    }
```

```java
        5 usages
15      @Autowired
16      private EmployeeRepository repo;
17
        no usages
18      @PostMapping("/add")
19      public Employee addEmployee(@RequestBody Employee emp) {
20          return repo.save(emp);
21      }
22
        no usages
23      @GetMapping("/all")
24      public List<Employee> all() {
25          return repo.findAll();
26      }
27
        no usages
28      @GetMapping("/{id}")
29      public Optional<Employee> byId(@PathVariable String id) {
30          return repo.findById(id);
31      }
32
        no usages
33      @GetMapping("/byEmployeeId/{employeeId}")
34      public Optional<Employee> byEmployeeId(@PathVariable String employeeId) {
35          return repo.findByEmployeeId(employeeId);
36      }
37
```

```java
1      package com.example.employee_api.controller;
2
3      import com.example.employee_api.model.Employee;
4      import com.example.employee_api.repository.EmployeeRepository;
5      import org.springframework.beans.factory.annotation.Autowired;
6      import org.springframework.stereotype.Controller;
7      import org.springframework.ui.Model;
8      import org.springframework.web.bind.annotation.*;
9
       no usages
10     @Controller
11     @RequestMapping("/employee")
12     public class WebController {
13
          6 usages
14         @Autowired
15         private EmployeeRepository employeeRepository;
16
17         // Show form (index page)
           no usages
18         @GetMapping("/")
19 @       public String index(Model model) {
20             model.addAttribute( attributeName: "employee", new Employee());
21             return "index"; // index.html
22         }
23
24         // Save employee, then redirect (Post/Redirect/Get pattern)
           no usages
25         @PostMapping("/save")
26         public String saveEmployee(@ModelAttribute Employee employee) {
```

```java
            employeeRepository.save(employee);
            return "redirect:/employee/displayAll";
        }

        // Show all employees
        no usages
        @GetMapping("/displayAll")
        public String displayAll(Model model) {
            model.addAttribute( attributeName: "employees", employeeRepository.findAll());
            return "displayAll"; // displayAll.html
        }

        // Show details for an identifier (first try Mongo _id, then employeeId)
        no usages
        @GetMapping("/display/{identifier}")
        public String displayByIdentifier(@PathVariable String identifier, Model model) {
            Employee emp = employeeRepository.findById(identifier).orElse( other: null);
            if (emp == null) {
                emp = employeeRepository.findByEmployeeId(identifier).orElse( other: null);
            }
            model.addAttribute( attributeName: "employee", emp);
            return "displayById"; // ✅ matches your template name
        }

        // Search helper (form redirects here)
        no usages
        @GetMapping("/display/search")
        public String searchRedirect(@RequestParam("q") String q) {
            return "redirect:/employee/display/" + q;
        }
```

```java
            if (emp == null) {
                emp = employeeRepository.findByEmployeeId(identifier).orElse( other: null);
            }
            model.addAttribute( attributeName: "employee", emp);
            return "displayById"; // ✅ matches your template name
        }

        // Search helper (form redirects here)
        no usages
        @GetMapping("/display/search")
        public String searchRedirect(@RequestParam("q") String q) {
            return "redirect:/employee/display/" + q;
        }

        // Delete by Mongo _id
        no usages
        @GetMapping("/delete/{id}")
        public String deleteById(@PathVariable String id) {
            if (employeeRepository.existsById(id)) {
                employeeRepository.deleteById(id);
            }
            return "redirect:/employee/displayAll";
        }
    }
```

```java
package com.example.employee_api.model;


import org.springframework.data.annotation.Id;
import org.springframework.data.mongodb.core.mapping.Document;


13 usages
@Document(collection = "employee")
public class Employee {


    2 usages
    @Id
    private String id;           // MongoDB _id (auto-generated if not set)


    3 usages
    private String employeeId;   // Custom employee ID (entered on form)
    3 usages
    private String name;
    3 usages
    private String email;
    3 usages
    private String location;


    // Constructors
    11 usages
    public Employee() {}


    10 usages
    public Employee(String employeeId, String name, String email, String location) {
        this.employeeId = employeeId;
```

```java
    public Employee(String employeeId, String name, String email, String location) {
        this.employeeId = employeeId;
        this.name = name;
        this.email = email;
        this.location = location;
    }

    // Getters & setters
    no usages
    public String getId() { return id; }
    no usages
    public void setId(String id) { this.id = id; }


    no usages
    public String getEmployeeId() { return employeeId; }
    no usages
    public void setEmployeeId(String employeeId) { this.employeeId = employeeId; }

    public String getName() { return name; }
    public void setName(String name) { this.name = name; }

    no usages
    public String getEmail() { return email; }
    no usages
    public void setEmail(String email) { this.email = email; }

    no usages
    public String getLocation() { return location; }
    no usages
    public void setLocation(String location) { this.location = location; }
```

```xml
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>3.5.5</version>
        <relativePath/> <!-- lookup parent from repository -->
    </parent>
    <groupId>com.example</groupId>
    <artifactId>employee-api</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <name>employee-api</name>
    <description>Demo project for Spring Boot</description>
    <url/>
    <licenses>
        <license/>
    </licenses>
    <developers>
        <developer/>
    </developers>
    <scm>
        <connection/>
        <developerConnection/>
        <tag/>
        <url/>
    </scm>
    <properties>
```

```xml
    </scm>
    <properties>
        <java.version>17</java.version>
    </properties>
    <dependencies>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-data-mongodb</artifactId>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-thymeleaf</artifactId>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-web</artifactId>
        </dependency>

        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-test</artifactId>
            <scope>test</scope>
        </dependency>
    </dependencies>

    <build>
        <plugins>
            <plugin>
                <groupId>org.springframework.boot</groupId>
```

```
40              </dependency>
41              <dependency>
42                  <groupId>org.springframework.boot</groupId>
43                  <artifactId>spring-boot-starter-web</artifactId>
44              </dependency>
45
46              <dependency>
47                  <groupId>org.springframework.boot</groupId>
48                  <artifactId>spring-boot-starter-test</artifactId>
49                  <scope>test</scope>
50              </dependency>
51          </dependencies>
52
53          <build>
54              <plugins>
55                  <plugin>
56                      <groupId>org.springframework.boot</groupId>
57                      <artifactId>spring-boot-maven-plugin</artifactId>
58                  </plugin>
59              </plugins>
60          </build>
61
62      </project>
63
```

```
1   package com.example.employee_api;
2
3 > import ...
5
6   @SpringBootTest
7   class EmployeeApiApplicationTests {
8
9       @Test
10      void contextLoads() {
11      }
12
13  }
14
```

EmployeeApiApplicationTests (com.example.employee_api) 1 sec 390 ms
    ✓ contextLoads()   1 sec 390 ms

✓ Tests passed: 1 of 1 test – 1 sec 390 ms

```
"C:\Program Files\Java\jdk-17.0.12\bin\java.exe" ...
12:49:31.531 [main] INFO org.springframework.test.context.
12:49:31.779 [main] INFO org.springframework.boot.test.co

  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/

 :: Spring Boot ::                (v3.5.5)

2025-09-08T12:49:32.863+05:30  INFO 22976 --- [employee-ap
2025-09-08T12:49:32.866+05:30  INFO 22976 --- [employee-ap
2025-09-08T12:49:34.118+05:30  INFO 22976 --- [employee-ap
2025-09-08T12:49:34.222+05:30  INFO 22976 --- [employee-ap
2025-09-08T12:49:35.321+05:30  INFO 22976 --- [employee-ap
2025-09-08T12:49:35.347+05:30  INFO 22976 --- [employee-ap
2025-09-08T12:49:36.244+05:30  INFO 22976 --- [employee-ap
2025-09-08T12:49:37.041+05:30  INFO 22976 --- [employee-ap
```

Creating the employee-db in mongodb

```
Microsoft Windows [Version 10.0.26100.4946]
(c) Microsoft Corporation. All rights reserved.

C:\Users\manju>mongosh
Current Mongosh Log ID: 68be729f9e561f96abeec4a8
Connecting to:          mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2
.5.6
Using MongoDB:          8.0.12
Using Mongosh:          2.5.6
mongosh 2.5.7 is available for download: https://www.mongodb.com/try/download/shell

For mongosh info see: https://www.mongodb.com/docs/mongodb-shell/

------
   The server generated these startup warnings when booting
   2025-09-07T20:11:20.776+05:30: Access control is not enabled for the database. Read and write access to data and conf
iguration is unrestricted
------

test> use employee_db
... db.createCollection("employee")
...
switched to db employee_db
employee_db>
```

Output:

displayAll the employee

displayById:

Employee List

Search by Employee ID or Mongo ID: [4] [Search]

---

localhost:8080/employee/display/4

Employee Details

Mongo _id: 68be80f6a8929448b29637ed
Employee ID: 4
Name: ella
Email: ella@gmail.com
Location: mumbai

Back to All | Add New