



PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013)

Object Oriented Analysis and Design using Java (UE20CS352)

Self Learning Hands-on Assignment: MVC Framework

Submitted by: Mahika Gupta
SRN: PES1UG20CS243

MVC Architecture Pattern:

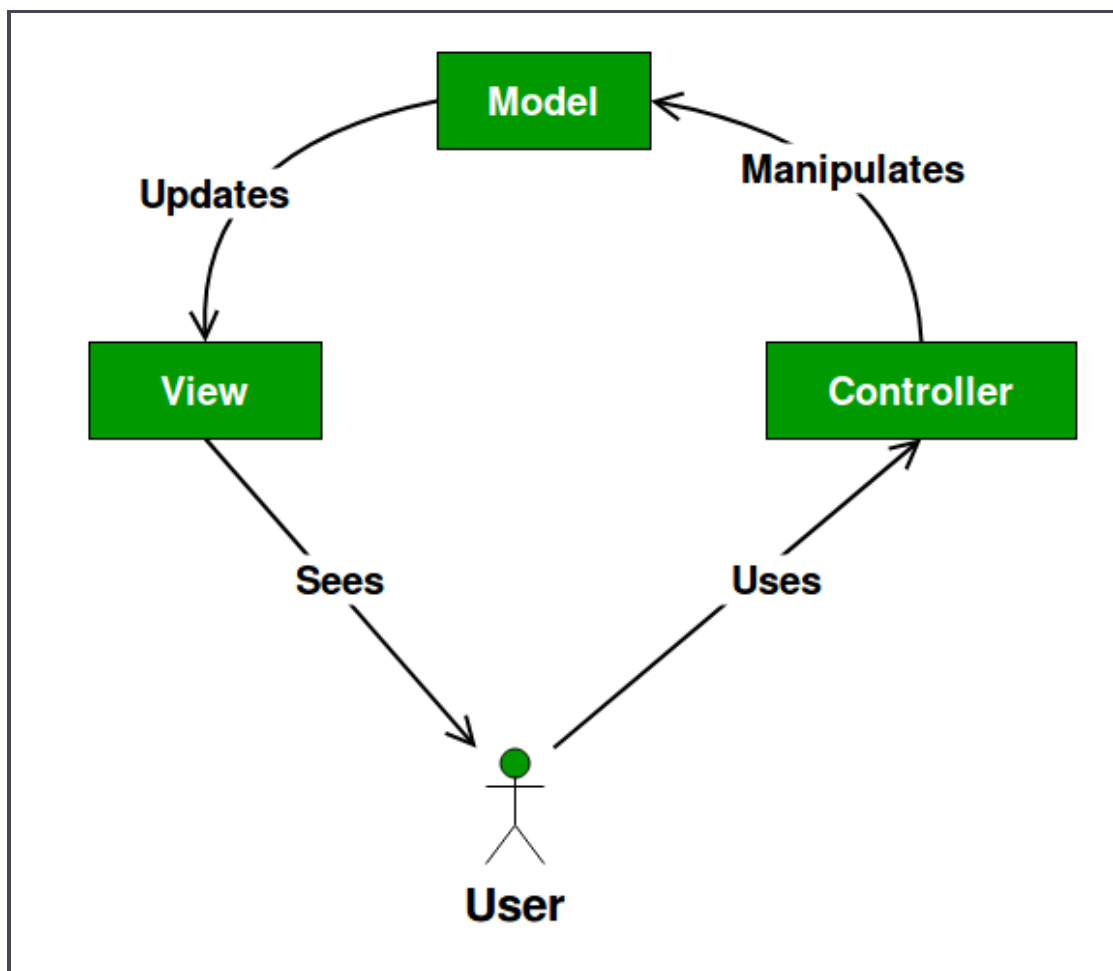
The Model-View-Controller (MVC) is an architectural pattern used in software engineering for designing and developing user interfaces. It separates the components of an application into three interconnected parts: the Model, the View, and the Controller.

In the MVC pattern:

Model: It is the core component that contains the business logic and data of the application. It interacts with the database and provides data to the controller.

View: is responsible for displaying data to the user and receives input from the user.

Controller: acts as the mediator between the model and the view. It receives input from the user via the view, interacts with the model to retrieve data, and sends the data to the view for display.



<https://media.geeksforgeeks.org/wp-content/uploads/MVC-Design-Pattern.png>

Advantages:

- MVC pattern provides a clear separation of components, resulting in a separation of concerns.
- Each component can be developed independently, simplifying the maintenance and modification of the application.
- The pattern enables the reusability of components in different applications, reducing development time and costs.
- The separation of concerns increases the testability of the application as each component can be tested individually.
- The MVC pattern allows for the scalability of the application through the addition or removal of components to meet changing business needs.
- The pattern enhances maintainability by enabling the modification of individual components without affecting the rest of the application, reducing the risk of errors and streamlining the problem-solving process.

Features of Chosen Frameworks:

Here are some features of Vaadin and Spring Boot frameworks with respect to MVC Architecture:

Vaadin:

- **Component-based architecture:** Vaadin is a component-based framework, where the user interface (UI) is built using pre-built components, reducing the amount of code needed to build an application.
- **Server-side architecture:** Vaadin utilizes a server-side architecture, where the components are rendered on the server and sent to the client as HTML, CSS, and JavaScript.
- **Data binding:** Vaadin supports two-way data binding, which allows changes made in the UI to be automatically reflected in the data model and vice versa.
- **Built-in testing tools:** Vaadin includes built-in testing tools, such as TestBench, which allows for functional testing of the UI components.

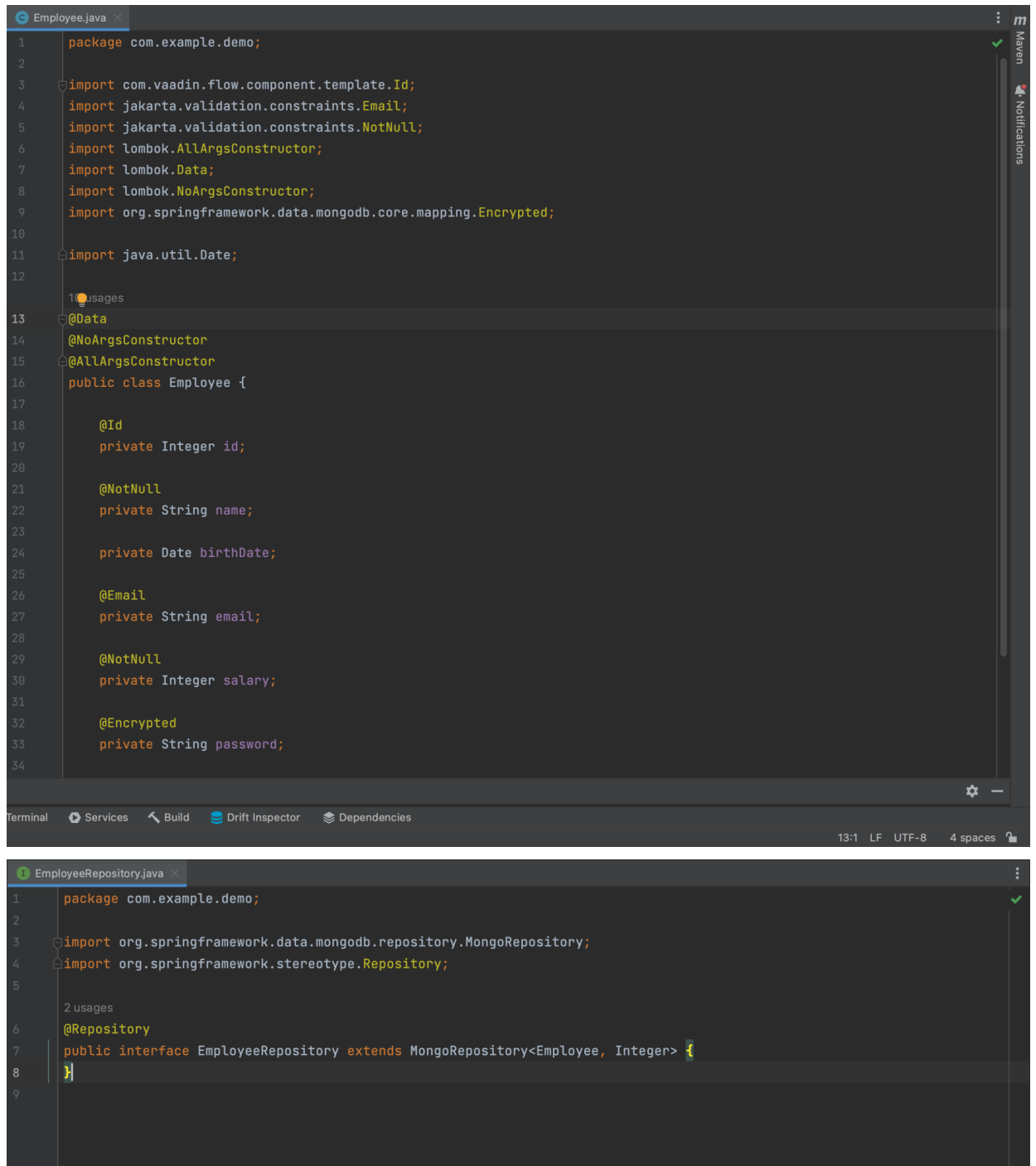
Spring Boot:

- **Convention over configuration:** Spring Boot follows the principle of convention over configuration, making it easy to set up a new project with minimal configuration.
- **Dependency Injection:** Spring Boot uses dependency injection, allowing for loose coupling between components and improving the maintainability of the application.
- **Thymeleaf:** Spring Boot supports Thymeleaf, a modern server-side Java template engine, which allows for the easy rendering of dynamic web pages.
- **RESTful web services:** Spring Boot includes built-in support for creating RESTful web services, making it easy to create APIs and microservices.

Problem Statement: Employee Management System

The problem statement for an employee management system is to create a centralized platform that can effectively manage employee data and streamline various HR processes. An employee management system should enable organizations to track employee information, such as personal details, salary, Employee ID, etc. With an employee management system, organizations can improve their HR processes, reduce paperwork, increase efficiency, and enhance employee satisfaction.

Model Class:



The image shows a screenshot of an IDE with two Java files open. The top file is `Employee.java` and the bottom file is `EmployeeRepository.java`.

Employee.java

```
1 package com.example.demo;
2
3 import com.vaadin.flow.component.template.Id;
4 import jakarta.validation.constraints.Email;
5 import jakarta.validation.constraints.NotNull;
6 import lombok.AllArgsConstructor;
7 import lombok.Data;
8 import lombok.NoArgsConstructor;
9 import org.springframework.data.mongodb.core.mapping.Encrypted;
10
11 import java.util.Date;
12
13 @Data
14 @NoArgsConstructor
15 @AllArgsConstructor
16 public class Employee {
17
18     @Id
19     private Integer id;
20
21     @NotNull
22     private String name;
23
24     private Date birthDate;
25
26     @Email
27     private String email;
28
29     @NotNull
30     private Integer salary;
31
32     @Encrypted
33     private String password;
34 }
```

EmployeeRepository.java

```
1 package com.example.demo;
2
3 import org.springframework.data.mongodb.repository.MongoRepository;
4 import org.springframework.stereotype.Repository;
5
6 @Repository
7 public interface EmployeeRepository extends MongoRepository<Employee, Integer> {
8
9 }
```

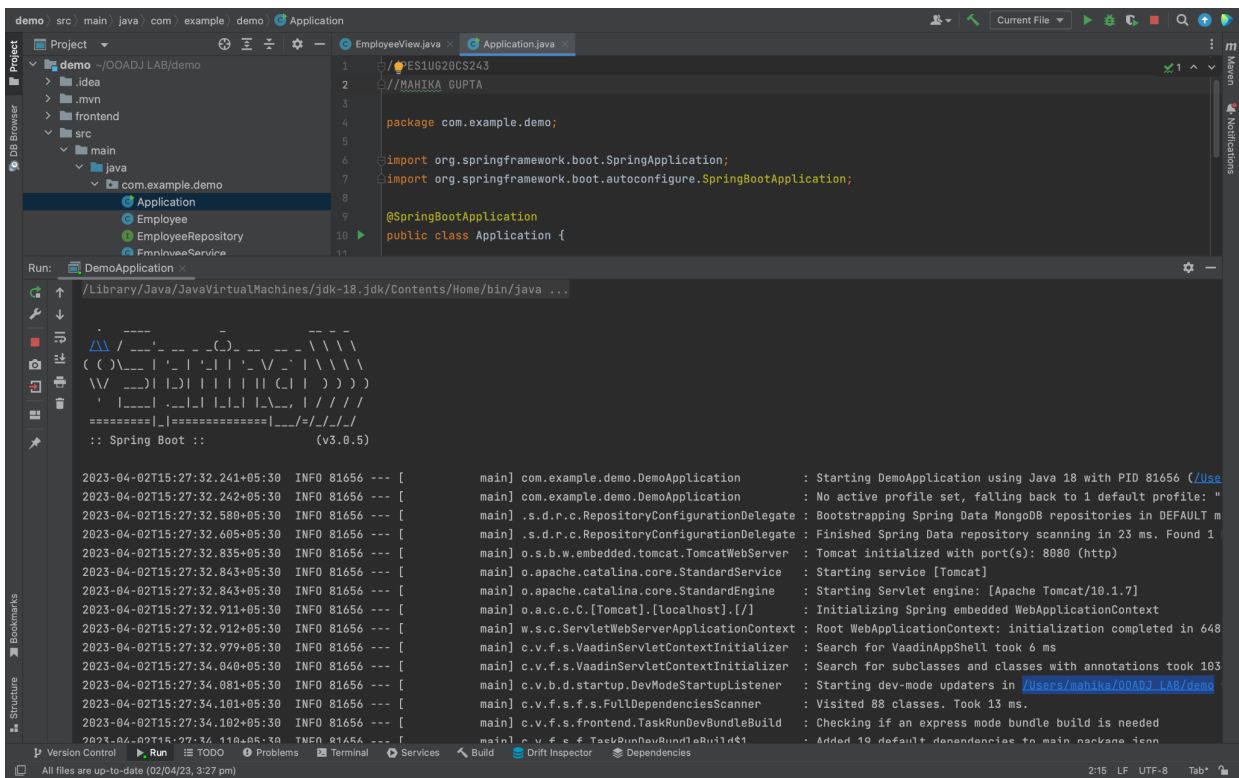
Controller Class:

```
EmployeeService.java
1 package com.example.demo;
2 import java.util.*;
3
4 import org.springframework.stereotype.Service;
5 import org.springframework.stereotype.Repository;
6 import org.vaadin.crudui.crud.CrudListener;
7
8 1 usage
9 @Service
10 public class EmployeeService implements CrudListener<Employee>{
11     5 usages
12     private final EmployeeRepository repository;
13
14     no usages
15     public EmployeeService(EmployeeRepository repository) { this.repository = repository; }
16
17     @Override
18     public List<Employee> findAll() { return repository.findAll(); }
19
20
21     @Override
22     public Employee add(Employee employee) { return repository.insert(employee); }
23
24
25
26     @Override
27     public Employee update(Employee employee) { return repository.save(employee); }
28
29
30
31     @Override
32     public void delete(Employee employee) { repository.delete(employee); }
33
34 }
35
36
```

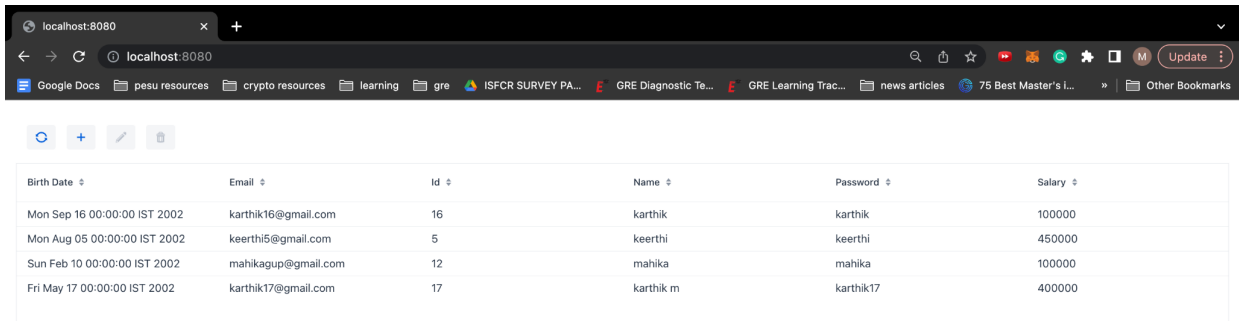
View Class:

```
EmployeeView.java
1 package com.example.demo;
2
3 import com.vaadin.flow.component.grid.Grid;
4 import com.vaadin.flow.component.orderedlayout.VerticalLayout;
5 import com.vaadin.flow.router.Route;
6 import org.vaadin.crudui.crud.impl.GridCrud;
7
8 no usages
9 @Route("")
10 public class EmployeeView extends VerticalLayout {
11     no usages
12     public EmployeeView(EmployeeService service) {
13         GridCrud<Employee> crud = new GridCrud<>(Employee.class, service);
14         add(crud);
15         setSizeFull();
16     }
17
18 }
```

Screenshot of console with application running



- Screenshot of UIs related to the two scenarios with values, outputs, errors (if any)



Inserting a new value:

localhost:8080

localhost:8080

Google Docspesu resourcescrypto resourceslearninggreISFCR SURVEY PA...GRE Diagnostic Te...GRE Learning Trac...news articles75 Best Master's I...Other Bookmarks

Birth Date

Email

Id

Name

Password

Salary

Mon Sep 16 00:00:00 IST 2002	karthik16@gmail.com	16	karthik	karthik	100000
Mon Aug 05 00:00:00 IST 2002	keerthi5@gmail.com	5	keerthi	keerthi	450000
Sun Feb 10 00:00:00 IST 2002	mahikagup@gmail.com	12	mahika	mahika	100000
Fri May 17 00:00:00 IST 2002	karthik17@gmail.com	17	karthik m	karthik17	400000

Add

Birth Date

17/6/2002

Email

mkm@gmail.com

Id

7

Name

maresh

Password

maresh123

Salary

430,000

✓ Add

Cancel

localhost:8080

localhost:8080

Google Docspesu resourcescrypto resourceslearninggreISFCR SURVEY PA...GRE Diagnostic Te...GRE Learning Trac...news articles75 Best Master's I...Other Bookmarks

Birth Date

Email

Id

Name

Password

Salary

Mon Sep 16 00:00:00 IST 2002	karthik16@gmail.com	16	karthik	karthik	100000
Mon Aug 05 00:00:00 IST 2002	keerthi5@gmail.com	5	keerthi	keerthi	450000
Sun Feb 10 00:00:00 IST 2002	mahikagup@gmail.com	12	mahika	mahika	100000
Fri May 17 00:00:00 IST 2002	karthik17@gmail.com	17	karthik m	karthik17	400000
Mon Jun 17 00:00:00 IST 2002	mkm@gmail.com	7	maresh	maresh123	430000

localhost:27017

Documents
test.employee

+

My Queries

Databases

Search

admin

config

local

mydb

test

employee

test.employee

31
DOCUMENTSINDEXES

DocumentsAggregationsSchemaExplain PlanIndexesValidation

Filter

Type a query: { field: 'value' }

ResetFindMore Options

ADD DATA

EXPORT COLLECTION

1 - 5 of 5

_id: 16

name: "karthik"

birthDate: 2002-09-15T18:30:00.000+00:00

email: "karthik16@gmail.com"

salary: 100000

password: "karthik"

_class: "com.example.demo.Employee"

_id: 5

name: "keerthi"

birthDate: 2002-08-04T18:30:00.000+00:00

email: "keerthi5@gmail.com"

salary: 450000

password: "keerthi"

_class: "com.example.demo.Employee"

_id: 12

name: "mahika"

birthDate: 2002-02-09T18:30:00.000+00:00

email: "mahikagup@gmail.com"

salary: 100000

password: "mahika"

_class: "com.example.demo.Employee"

_id: 17

name: "karthik m"

birthDate: 2002-05-16T18:30:00.000+00:00

email: "karthik17@gmail.com"

>_MONGOSH

Updating value:

localhost:8080

localhost:8080

Google Docspesu resourcescrypto resourceslearninggreISFCR SURVEY PA...GRE Diagnostic Te...GRE Learning Trac...news articles75 Best Master's L...Other Bookmarks

Birth Date	Email	Id	Name	Password	Salary
Mon Sep 16 00:00:00 IST 2002	karthik16@gmail.com	16	karthik	karthik	100000
Mon Aug 05 00:00:00 IST 2002	keerthi5@gmail.com	5	keerthi	keerthi	450000
Sun Feb 10 00:00:00 IST 2002	mahikagup@gmail.com	12	mahika	mahika	100000
Fri May 17 00:00:00 IST 2002	karthik17@gmail.com	17	karthik m	karthik17	400000
Mon Jun 17 00:00:00 IST 2002					000

Update

Birth Date

10/2/2002

Email

mahikagup@gmail.com

Id

10

Name

mahika

Password

mahika

Salary

100,000

Update

Cancel

localhost:27017

Documents
test.employee

My Queries

Databases

Search

admin

config

local

mydb

test

employee

test.employee

Documents

Aggregations

Schema

Explain Plan

Indexes

Validation

Filter

Type a query: { field: 'value' }

Reset

Find

More Options

ADD DATA

EXPORT COLLECTION

1 - 5 of 5

salary: 450000

password: "keerthi"

_class: "com.example.demo.Employee"

_id: 12

name: "mahika"

birthDate: 2002-02-09T18:30:00.000+00:00

email: "mahikagup@gmail.com"

salary: 100000

password: "mahika"

_class: "com.example.demo.Employee"

_id: 17

name: "karthik m"

birthDate: 2002-05-16T18:30:00.000+00:00

email: "karthik17@gmail.com"

salary: 400000

password: "karthik17"

_class: "com.example.demo.Employee"

_id: 7

name: "mahesh"

birthDate: 2002-06-16T18:30:00.000+00:00

email: "mkm@gmail.com"

salary: 430000

password: "mahesh123"

_class: "com.example.demo.Employee"

>_MONGOSH

localhost:8080

localhost:8080

Google Docspesu resourcescrypto resourceslearninggreISFCR SURVEY PA...GRE Diagnostic Te...GRE Learning Trac...news articles75 Best Master's L...Other Bookmarks

Birth Date	Email	Id	Name	Password	Salary
Mon Sep 16 00:00:00 IST 2002	karthik16@gmail.com	16	karthik	karthik	100000
Mon Aug 05 00:00:00 IST 2002	keerthi5@gmail.com	5	keerthi	keerthi	450000
Fri May 17 00:00:00 IST 2002	karthik17@gmail.com	17	karthik m	karthik17	400000
Mon Jun 17 00:00:00 IST 2002	mkm@gmail.com	7	mahesh	mahesh123	430000
Sun Feb 10 00:00:00 IST 2002	mahikagup@gmail.com	10	mahika	mahika	100000

Deleting value:

localhost:8080

localhost:8080

Google Docspesu resourcescrypto resourceslearninggreISFCR SURVEY PA...GRE Diagnostic Te...GRE Learning Trac...news articles75 Best Master's L...Other Bookmarks

Birth Date	Email	Id	Name	Password	Salary
Mon Sep 16 00:00:00 IST 2002	karthik16@gmail.com	16	karthik	karthik	100000
Mon Aug 05 00:00:00 IST 2002	keerthi5@gmail.com	5	keerthi	keerthi	450000
Fri May 17 00:00:00 IST 2002	karthik17@gmail.com	17	karthik m	karthik17	400000
Mon Jun 17 00:00:00 IST 2002	mkm@gmail.com	7	maresh	maresh123	430000
Sun Feb 10 00:00:00 IST 2002					100000

Are you sure you want to delete this item?

Birth Date

10/2/2002

Email

mahikagup@gmail.com

Id

10

Name

mahika

Password

mahika

Salary

100,000

Yes, delete

Cancel

localhost:8080

localhost:8080

Google Docspesu resourcescrypto resourceslearninggreISFCR SURVEY PA...GRE Diagnostic Te...GRE Learning Trac...news articles75 Best Master's L...Other Bookmarks

Birth Date	Email	Id	Name	Password	Salary
Mon Sep 16 00:00:00 IST 2002	karthik16@gmail.com	16	karthik	karthik	100000
Mon Aug 05 00:00:00 IST 2002	keerthi5@gmail.com	5	keerthi	keerthi	450000
Fri May 17 00:00:00 IST 2002	karthik17@gmail.com	17	karthik m	karthik17	400000
Mon Jun 17 00:00:00 IST 2002	mkm@gmail.com	7	maresh	maresh123	430000

localhost:27017

Documents
test.employee

My Queries

Databases

Search

admin

config

local

mydb

test

employee

test.employee

3 DOCUMENTS1 INDEXES

Documents

Aggregations

Schema

Explain Plan

Indexes

Validation

Filter

Type a query: { field: 'value' }

ResetFindMore Options

ADD DATA

EXPORT COLLECTION

1 - 4 of 4

_id: 16

name: "karthik "

birthDate: 2002-09-15T18:30:00.000+00:00

email: "karthik16@gmail.com"

salary: 100000

password: "karthik"

_class: "com.example.demo.Employee"

_id: 5

name: "keerthi"

birthDate: 2002-08-04T18:30:00.000+00:00

email: "keerthi5@gmail.com"

salary: 450000

password: "keerthi"

_class: "com.example.demo.Employee"

_id: 17

name: "karthik m"

birthDate: 2002-05-16T18:30:00.000+00:00

email: "karthik17@gmail.com"

salary: 400000

password: "karthik17"

_class: "com.example.demo.Employee"

_id: 7

name: "maresh"

birthDate: 2002-06-16T18:30:00.000+00:00

email: "mkm@gmail.com"

> MONGODB

Screenshot of database with data items

localhost:27017

Documents
test.employee

+

My Queries

Databases

Search

admin

config

local

mydb

test

employee

test.employee

3 DOCUMENTS1 INDEXES

DocumentsAggregationsSchemaExplain PlanIndexesValidation

FilterType a query: { field: 'value' }

ResetFindMore Options

ADD DATAEXPORT COLLECTION

1 - 4 of 4

_id: 16

name: "karthik "

birthDate: 2002-09-15T18:30:00.000+00:00

email: "karthik16@gmail.com"

salary: 100000

password: "karthik"

_class: "com.example.demo.Employee"

5

name: "keerthi"

birthDate: 2002-08-04T18:30:00.000+00:00

email: "keerthi5@gmail.com"

salary: 450000

password: "keerthi"

_class: "com.example.demo.Employee"

17

name: "karthik m"

birthDate: 2002-05-16T18:30:00.000+00:00

email: "karthik17@gmail.com"

salary: 400000

password: "karthik17"

_class: "com.example.demo.Employee"

7

name: "mahesh"

birthDate: 2002-06-16T18:30:00.000+00:00

email: "mkm@gmail.com"