# INSTITUTE FOR ADVANCED COMPUTING AND SOFTWARE DEVELOPMENT (IACSD), AKURDI, PUNE

Documentation On

# ReadersHub

PG-DAC Sept 2023

**Submitted By:**

Group No: 26

| **Roll No.** | **Name** |
|---|---|
| 239036 | Jate Sunny Gajanan |
| 239119 | Zatale Mayur Bhaurao |

**Mrs. Sonali Mogal**
**Project Guide**

**Mr. Rohit Puranik**
**Center Coordinator**

## **ABSTRACT**

The Readershub is an innovative software application designed to facilitatethe reading of books in a user-friendly and efficient manner. In today's fast life, the need for streamlined processes is essential for both author and reader. This system aims to provide a comprehensive platform that enables reader to manage their book inventory, check review about their work, and ensures better work to do.

The project's primary objective is to create a user-centric software solution that addresses the challenges faced by readers that it is not possible to carry books everywhere . By incorporating features such as online reading, user authentication, and real-time availability of book, the Readershub aims to revolutionize the readers problems.

With an emphasis on intuitive user interfaces, accurate book, and good reading document, the Readershub promises to optimize online reading operations and enhance readers satisfaction. By offering a centralized hub for both author and reader to interact, the system aims to foster better communication, faster decision-making, and increased transparency throughout the entire reading and publishing process.

As we embark on the development of the Readershub, we envision a future where reading books online operate more efficiently and readers experience a hassle-free reading process. By delivering a software product that aligns with the day to day life needs while preparing for future advancements, this project aims to contribute to the growth and modernization of the traditional book reading style. Through careful planning, rigorous development, and a dedication to excellence, we are committed to delivering a robust and user-friendly Readershub that transforms the way books were read.

## **ACKNOWLEDGEMENT**

I take this occasion to thank God, almighty for blessing us with his grace and taking our endeavor to a successful culmination. I extend my sincere and heartfelt thanks to our esteemed guide, Mrs. Sonali Mogal for providing me with the right guidance and advice at the crucial juncture sand for showing me the right way. I extend my sincere thanks to our respected Centre Co-Ordinator Mr. Rohit Puranik, for allowing us to use the facilities available. I would like to thank the other faculty members also, at this occasion. Last but not the least, I would like to thank my friends and family for the support and encouragement they have given me during the course of our work.

Jate Sunny Gajanan (230941220074)
Zatale Mayur Bhaurao (230941220223)

## **INDEX**

# 1. INTRODUCTION

In an era defined by rapid urbanization and busy work culture, the books plays a pivotal role in providing individuals with the means to read their favorite areas. The process of reading book any time, however, has traditionally been associated with various challenges and complexities. These range from the time-consuming nature of physical visits to store, to the uncertainty of finding the right book at affordable price. To address these challenges and bring a paradigm shift to the book market, we introduce "ReadersHub," a innovative onlineplatform designed to streamline and simplify the process of reading book.

The book industry has been a cornerstone of modern society, providing individuals with a sense of freedom and connectivity. As technological advancements continue to reshape various sectors, the book industry stands poised for transformation. ReadersHub is a response to the growing demand for a digital ecosystem that not only connects Authors and Readers seamlessly, but also enhances the overall experience of reading different types of books. By leveraging the power of technology, data and user-centric design, ReadersHub to revolutionize the way of book reading.

## Purpose

The purpose of this document is to provide a detailed specification of the features and functionalities of the "Readershub" It outlines the requirements, system architecture, and user interactions.

## Scope

The system is designed to manage information related to book, author and reader documents. It facilitates tasks such as publish book, maintain readers reading history, book rating user management document uploads.

## Objective of Project on Readershub:

The Readershub project is designed with a clear set of objectives to enhance the book online reading process. The primary goal is to provide availability of multiple book for reading with a streamlined platform to efficiently manage their book.
Key objectives include creating a user-friendly experience for author and reader, ensuring secure user authentication for data protection.
The system focuses on maintaining accurate book information, enhancing data accuracy and integrity.
With modern technology and a future-ready approach, the project aims to create a reliable, transparent, and efficient platform that benefits to all book reading process.

## Functionalities provided by Readershub are as follows:

The "Readershub" project provides a range of functionalities aimed at simplifying and enhancing the book reading process for users. Some of the key functionalities offered by the project include:

Book Listings: Author can easily publish their books on the platform, providing detailed information about each book, including author, genre of book, language and book description.

User Registration and Authentication: Users can create accounts and securely log in to the platform. Secure authentication mechanisms help protect user data and ensure that only authorized individuals can access the system. If user is author then can publish the book and also can read books of different authors. If user is reader the can read one book at a time and give rating accordingly.
Search and Filtering: Reader can search for book based on their preferences, such as author, book name, genre and language. Advanced filtering options make it easy to narrow down search results.
Book details: Detailed information about each listed vehicle is provided, including high- quality pdf for online reading, specifications, features, and author information.
Review rating: The platform facilitates user to give rating for book according the book content
Reader Profiles: Reader can create and manage their profiles, including email and mobile no. information.
Author Dashboard: Author have access to a dashboard that allows them to publish their book, check rating given by users.Data Integrity: The system ensures good contents/books across the platform, reducing errors and

## REQUIREMENTS

### Functional Requirements

FR 1. User Registration and Authentication:
Users can register by providing necessary details.
Forgot password functionality allows users to reset their passwords.

FR 2. Book Management:
Authors can add, update, or delete book and details.
Book categorized in genre and language.
Books are uploaded in form of pdf, this pdf will open in online pdf viewer.

FR 3. Readers Profiles:
Reader can view and update their profiles.
Reader profiles store personal information.

FR 4. Author Profiles:
Author can view and update their profiles.
Author's profile store personal information.
Author can publish their book.

FR 7. Pdf Upload:
Author can upload PDF of their book.
Reader can view pdf of that book.

FR 8. Error Handling and Reporting:
The system handles errors gracefully and provides appropriate error messages.
Admins can access logs and error reports for troubleshooting.

### Non Functional Requirements:

NFR 1. Security:
User passwords are securely stored using encryption techniques.

NFR 2. Performance:
The system should handle a large number of simultaneous users without significantslowdowns.
PDF loading and retrieval should be efficient for a smooth user experience.

NFR 3. Scalability:
The system should be designed to accommodate future growth and increased user activity.

NFR 4. Usability:
The user interface should be intuitive and user-friendly for both reader/ author and administrator.
Clear and concise error messages should guide users through any issues.

NFR 5. Reliability

The system should be available and operational 24/7 with minimal downtime.

NFR 6. Data Integrity:
Data integrity and consistency are maintained through proper validation and database design.

NFR 7. Data Privacy:
User data, especially personal and sensitive information, should be stored securely.

## Hardware and Network Interfaces:

Back-end Server Configuration:
Intel core i5 Processor
128 MB RAM
1 Raid Controller Card
32-bit Ethernet Controller (100 Base-T)
8 x 2.0 GB Fast SCSI/2 with Raid Support
2.88 MB FDD
48x CD ROM Drive

SVGA Colour Monitor on PCI with 1MB RAM
101 Keys Keyboard
1 Microsoft Mouse with pad
4/8 GB DAT
One Serial & Two Parallel Ports
Internet Information Server (IIS)
Microsoft Transaction Server (MTS)

Front-end Client Configuration:

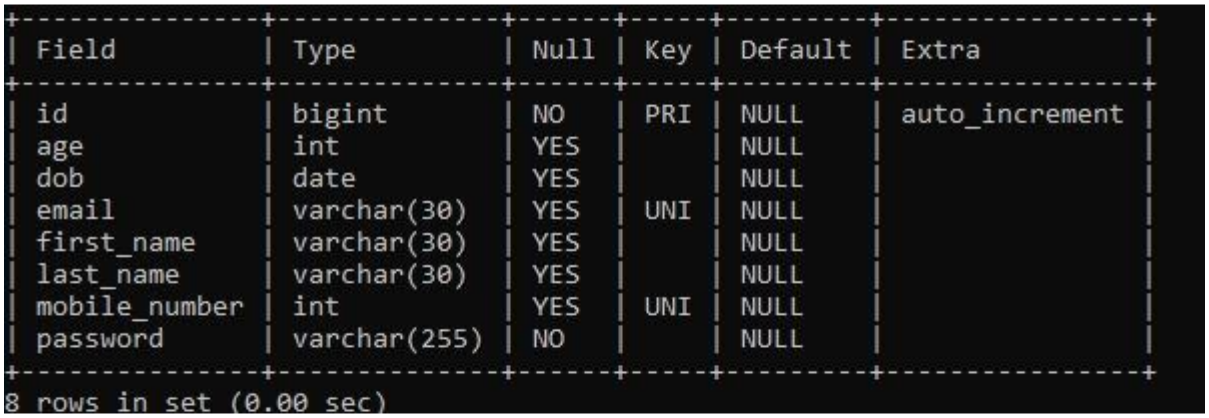
## Software Interfaces:

Software configuration for back-end Services:
STS, MySql WorkBench
Command Line Interface
Software configuration for front-end Services:
VS Code
Client Workstation
Office 2000
-Web Browser – Internet Explorer

## DATABASE DESIGN

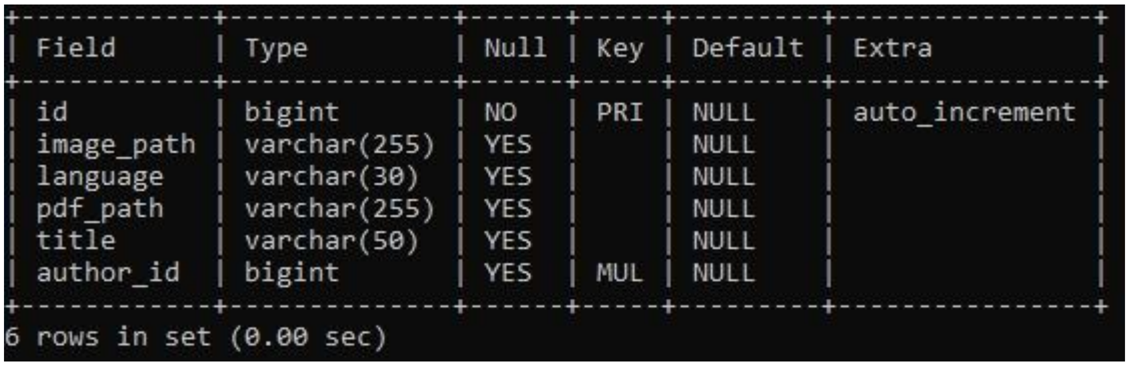
The following table structures depict the database design.

Table 1: Author:



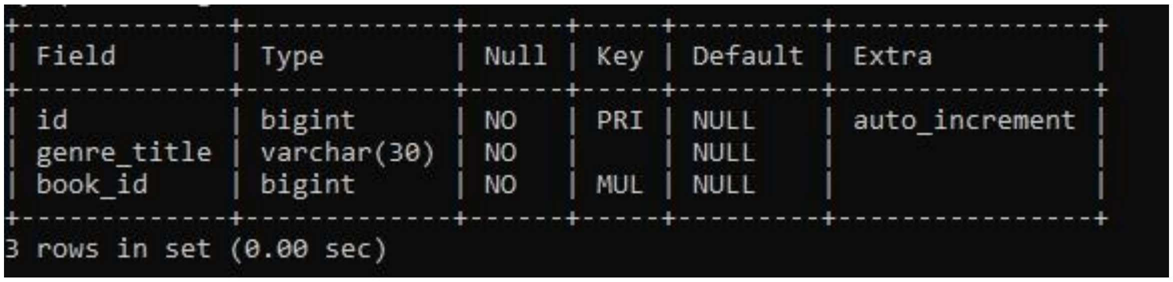
```
+---------------+--------------+------+-----+---------+----------------+
| Field         | Type         | Null | Key | Default | Extra          |
+---------------+--------------+------+-----+---------+----------------+
| id            | bigint       | NO   | PRI | NULL    | auto_increment |
| age           | int          | YES  |     | NULL    |                |
| dob           | date         | YES  |     | NULL    |                |
| email         | varchar(30)  | YES  | UNI | NULL    |                |
| first_name    | varchar(30)  | YES  |     | NULL    |                |
| last_name     | varchar(30)  | YES  |     | NULL    |                |
| mobile_number | int          | YES  | UNI | NULL    |                |
| password      | varchar(255) | NO   |     | NULL    |                |
+---------------+--------------+------+-----+---------+----------------+
8 rows in set (0.00 sec)
```

Table 2: Books:



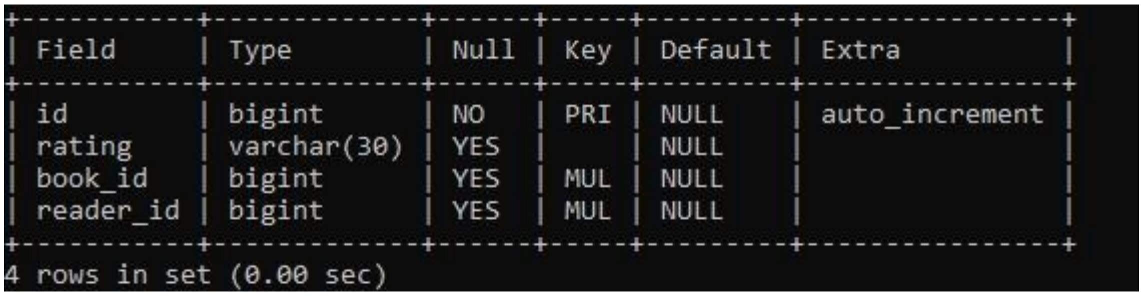
```
+------------+--------------+------+-----+---------+----------------+
| Field      | Type         | Null | Key | Default | Extra          |
+------------+--------------+------+-----+---------+----------------+
| id         | bigint       | NO   | PRI | NULL    | auto_increment |
| image_path | varchar(255) | YES  |     | NULL    |                |
| language   | varchar(30)  | YES  |     | NULL    |                |
| pdf_path   | varchar(255) | YES  |     | NULL    |                |
| title      | varchar(50)  | YES  |     | NULL    |                |
| author_id  | bigint       | YES  | MUL | NULL    |                |
+------------+--------------+------+-----+---------+----------------+
6 rows in set (0.00 sec)
```

Table 3: Genre:



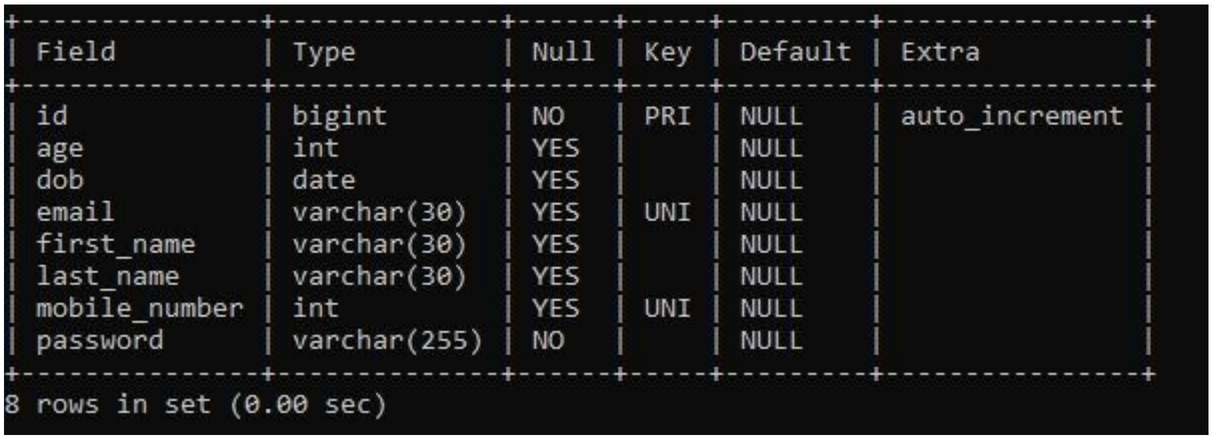
```
+-------------+-------------+------+-----+---------+----------------+
| Field       | Type        | Null | Key | Default | Extra          |
+-------------+-------------+------+-----+---------+----------------+
| id          | bigint      | NO   | PRI | NULL    | auto_increment |
| genre_title | varchar(30) | NO   |     | NULL    |                |
| book_id     | bigint      | NO   | MUL | NULL    |                |
+-------------+-------------+------+-----+---------+----------------+
3 rows in set (0.00 sec)
```

Table 4: Reader-book-log:



```
+-----------+-------------+------+-----+---------+----------------+
| Field     | Type        | Null | Key | Default | Extra          |
+-----------+-------------+------+-----+---------+----------------+
| id        | bigint      | NO   | PRI | NULL    | auto_increment |
| rating    | varchar(30) | YES  |     | NULL    |                |
| book_id   | bigint      | YES  | MUL | NULL    |                |
| reader_id | bigint      | YES  | MUL | NULL    |                |
+-----------+-------------+------+-----+---------+----------------+
4 rows in set (0.00 sec)
```

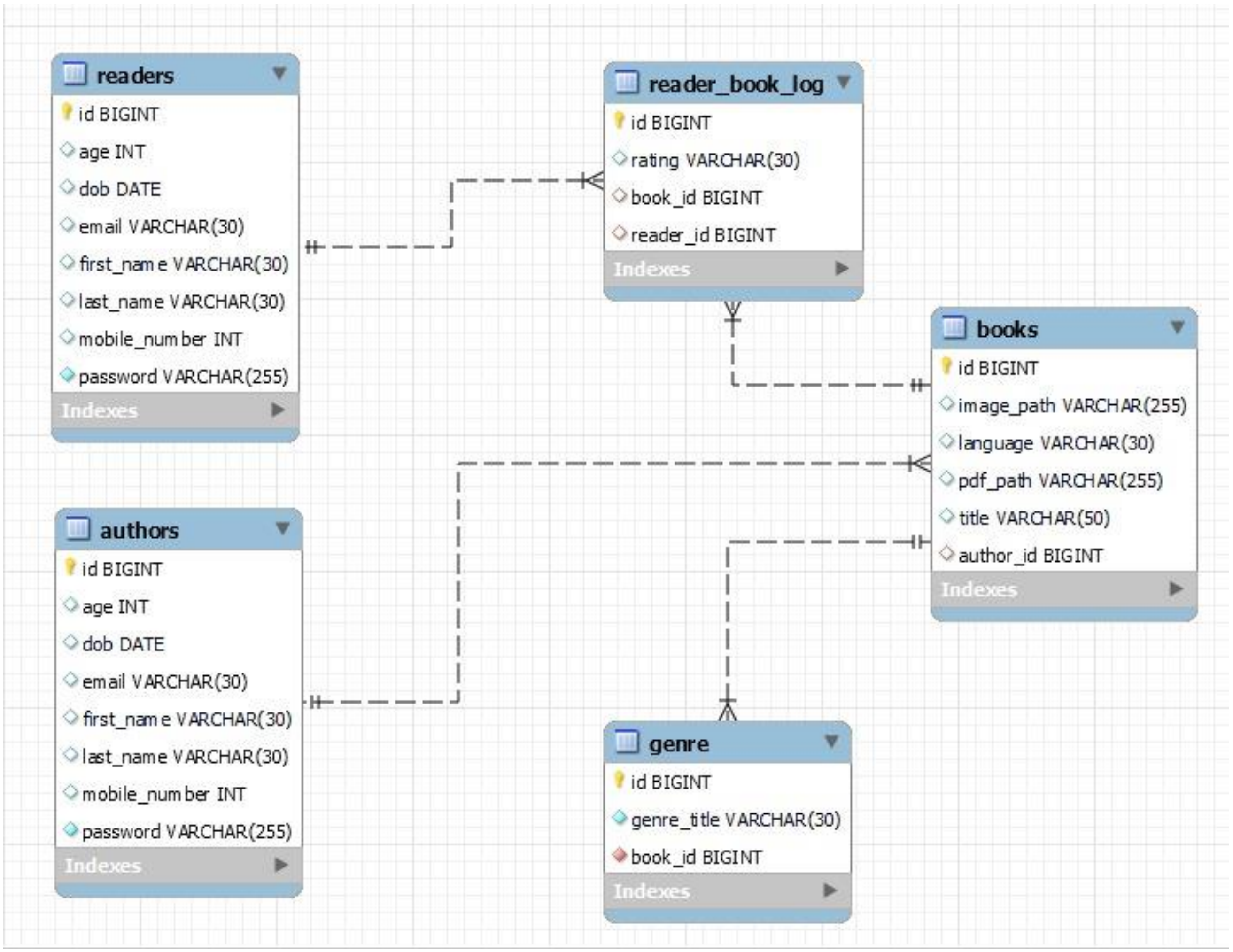
Table 5: Readers:

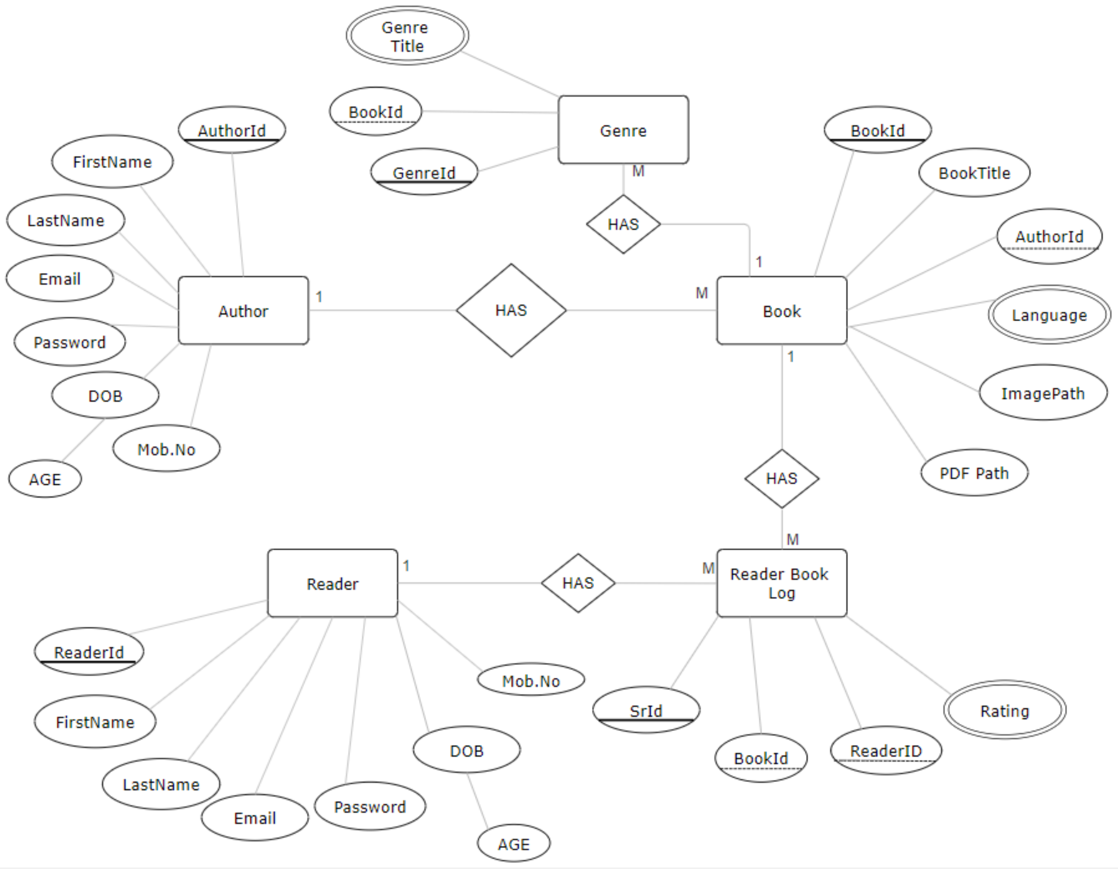


```
+---------------+--------------+------+-----+---------+----------------+
| Field         | Type         | Null | Key | Default | Extra          |
+---------------+--------------+------+-----+---------+----------------+
| id            | bigint       | NO   | PRI | NULL    | auto_increment |
| age           | int          | YES  |     | NULL    |                |
| dob           | date         | YES  |     | NULL    |                |
| email         | varchar(30)  | YES  | UNI | NULL    |                |
| first_name    | varchar(30)  | YES  |     | NULL    |                |
| last_name     | varchar(30)  | YES  |     | NULL    |                |
| mobile_number | int          | YES  | UNI | NULL    |                |
| password      | varchar(255) | NO   |     | NULL    |                |
+---------------+--------------+------+-----+---------+----------------+
8 rows in set (0.00 sec)
```

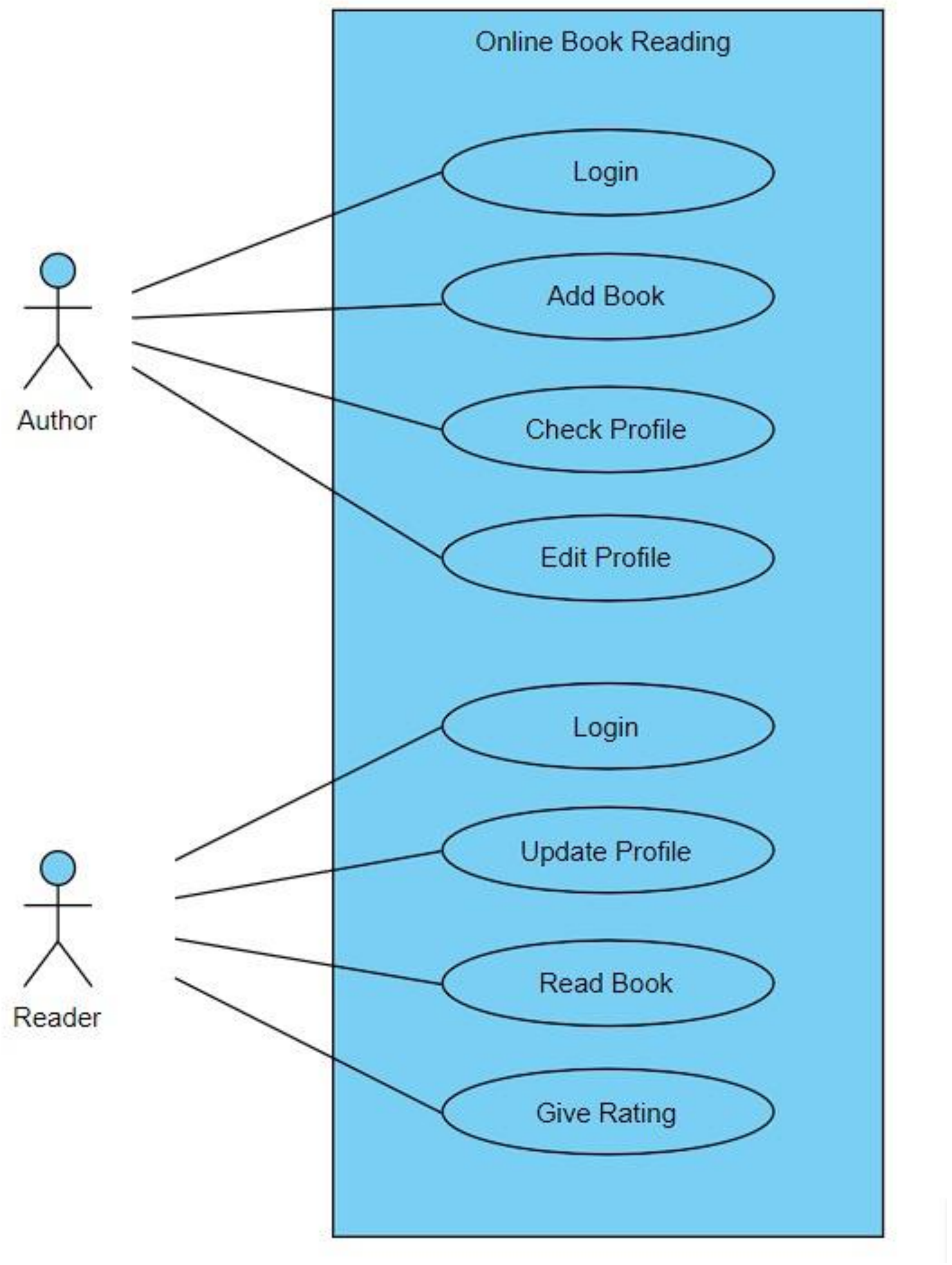
## APPENDIX A

**Entity Relationship Diagram:**



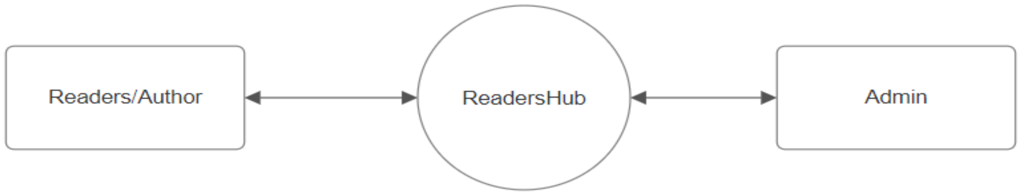

Genre Title
BookId
GenreId
Genre
M
HAS
1
AuthorId
FirstName
LastName
Email
Password
DOB
AGE
Mob.No
Author
1
HAS
M
Book
BookId
BookTitle
AuthorId
Language
ImagePath
PDF Path
1
HAS
M
Reader
1
HAS
M
Reader Book Log
ReaderId
FirstName
LastName
Email
Password
DOB
AGE
Mob.No
SrId
BookId
ReaderID
Rating

**Use Case Diagram:**

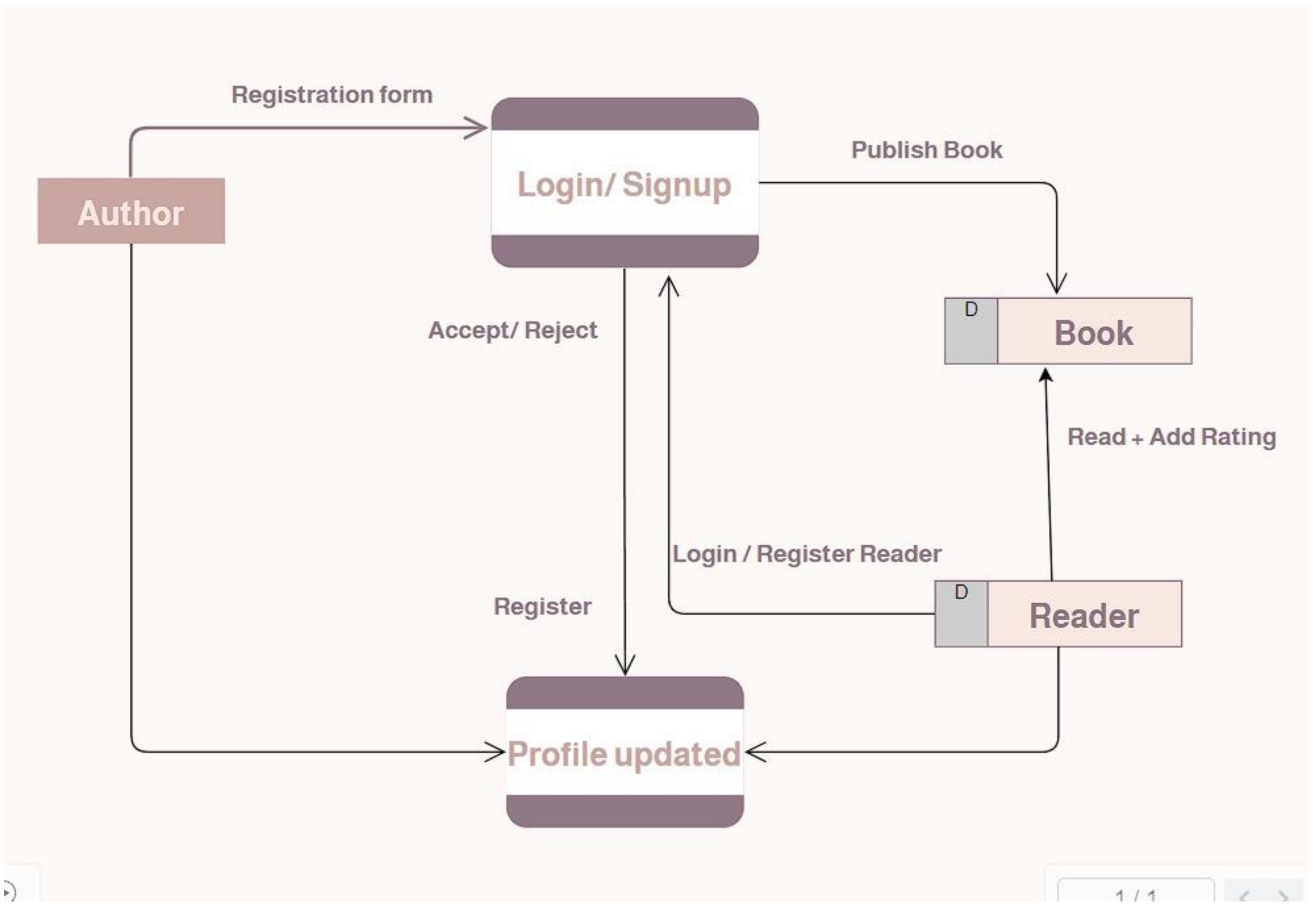


## Data Flow Diagram:

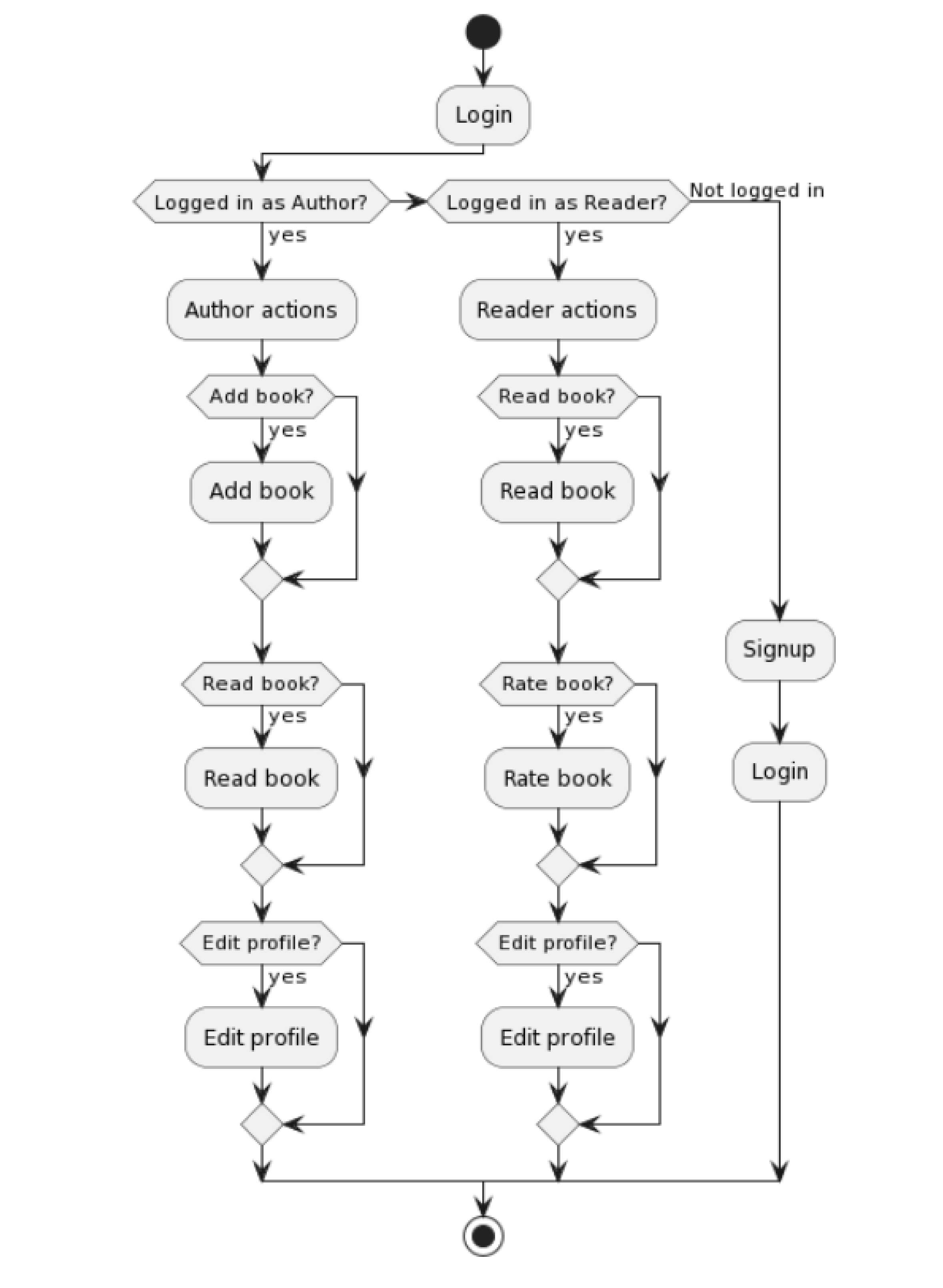
### Level 0:



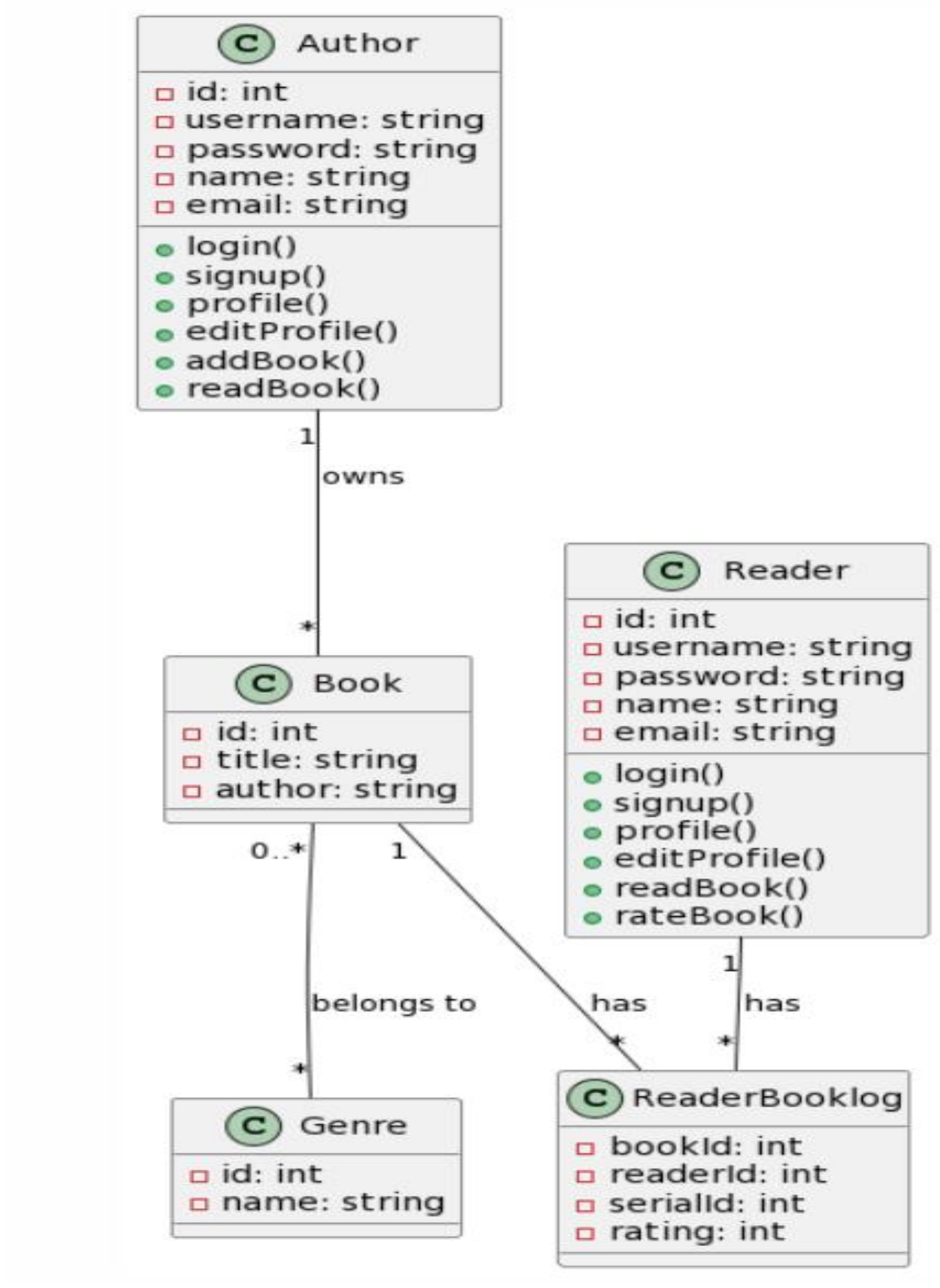
### level 1 :



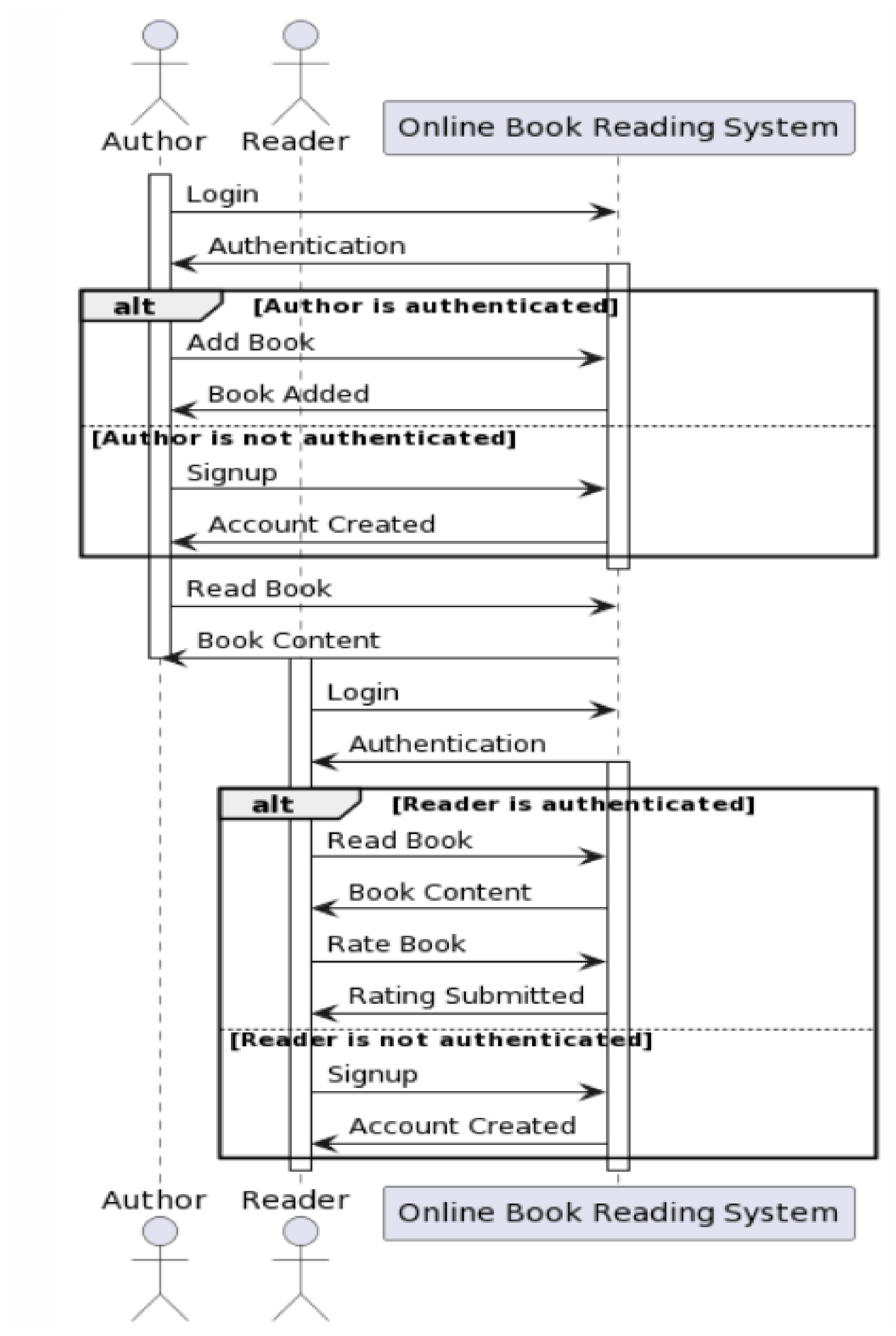
## Activity Diagram :



**Class diagram :**

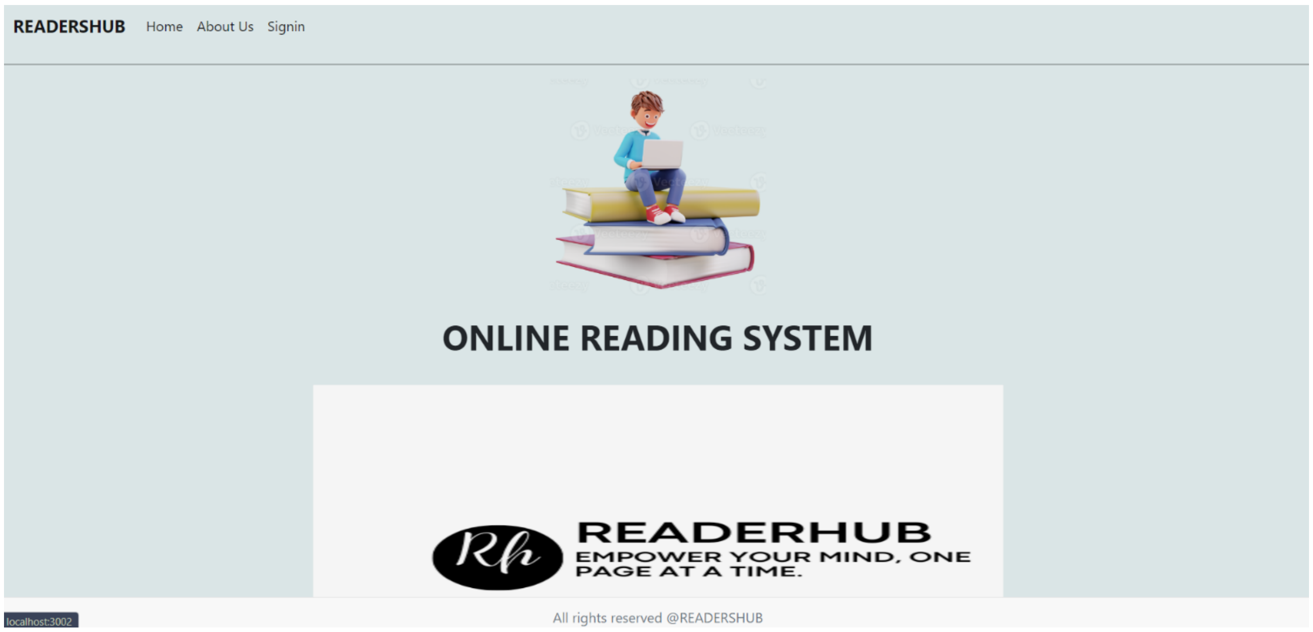


**Sequence diagram :**

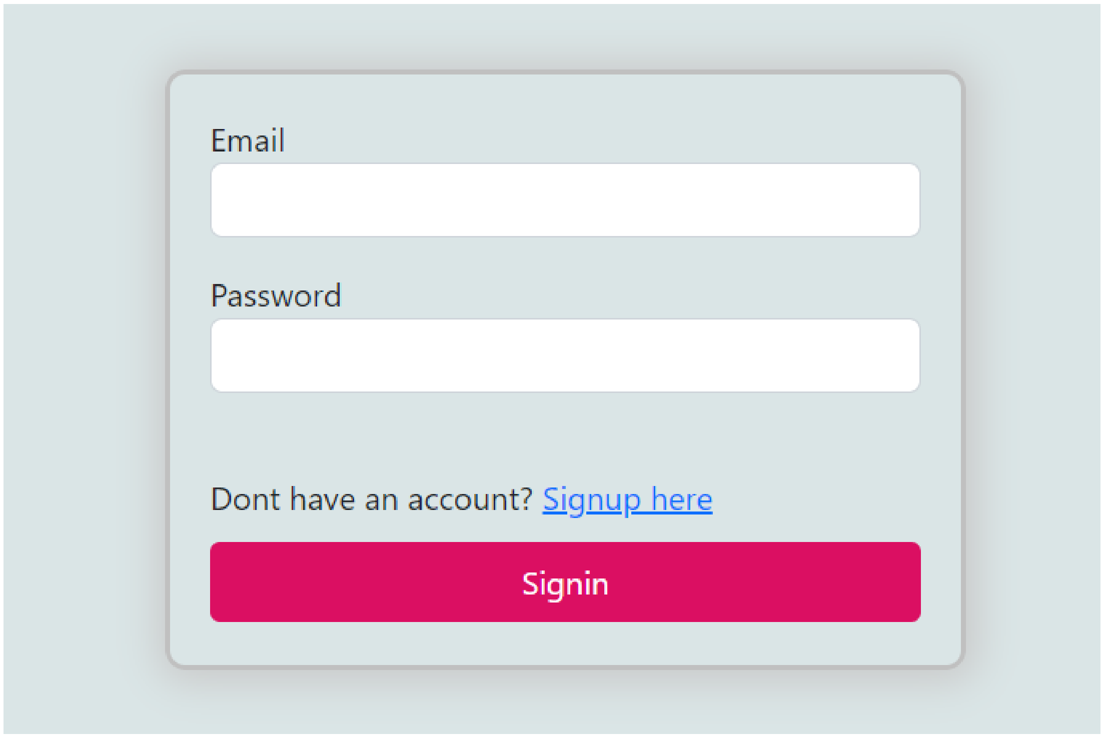


## APPENDIX B

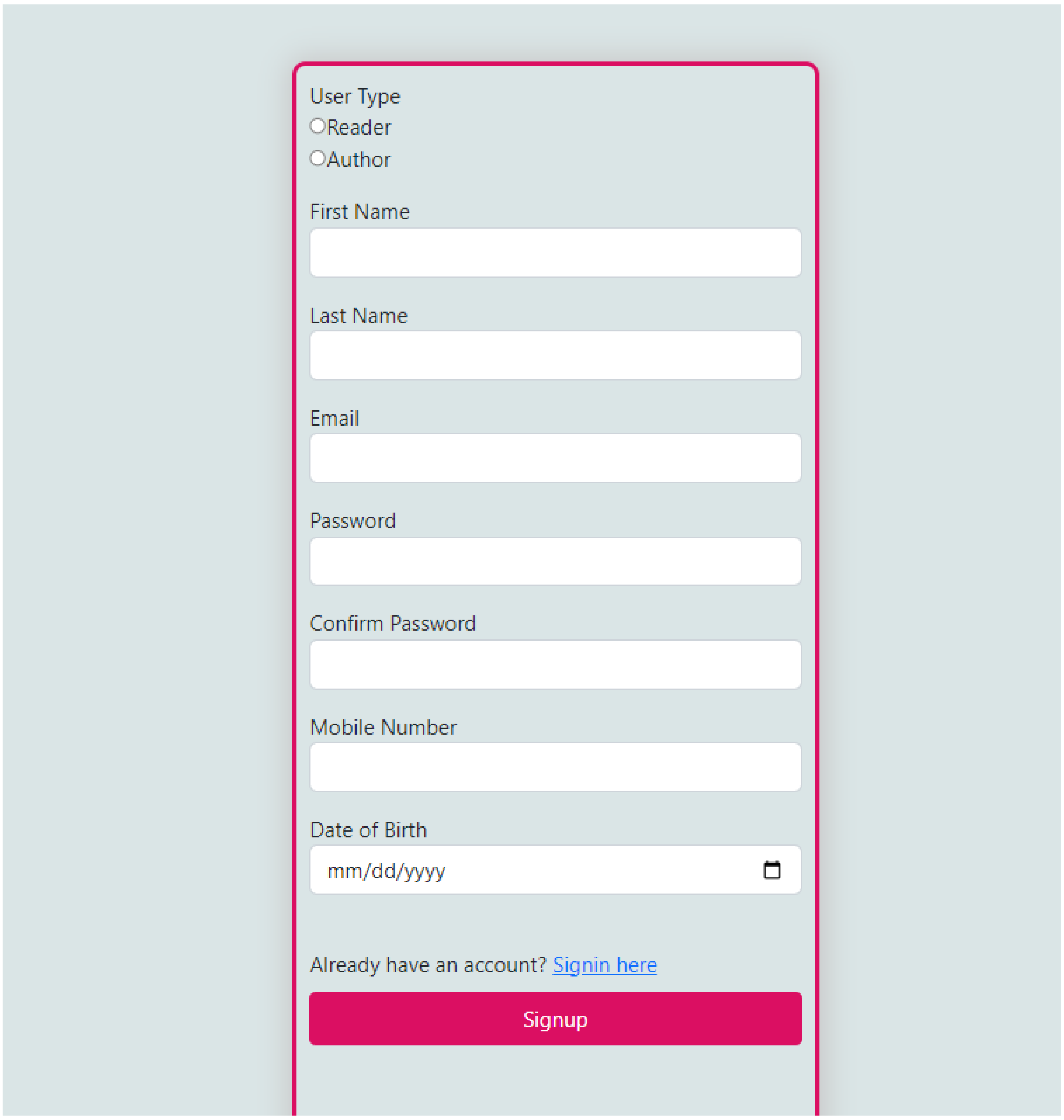
### Homepage:



### Login:



## New User signup



User Type

○Reader

○Author

First Name

Last Name

Email

Password

Confirm Password

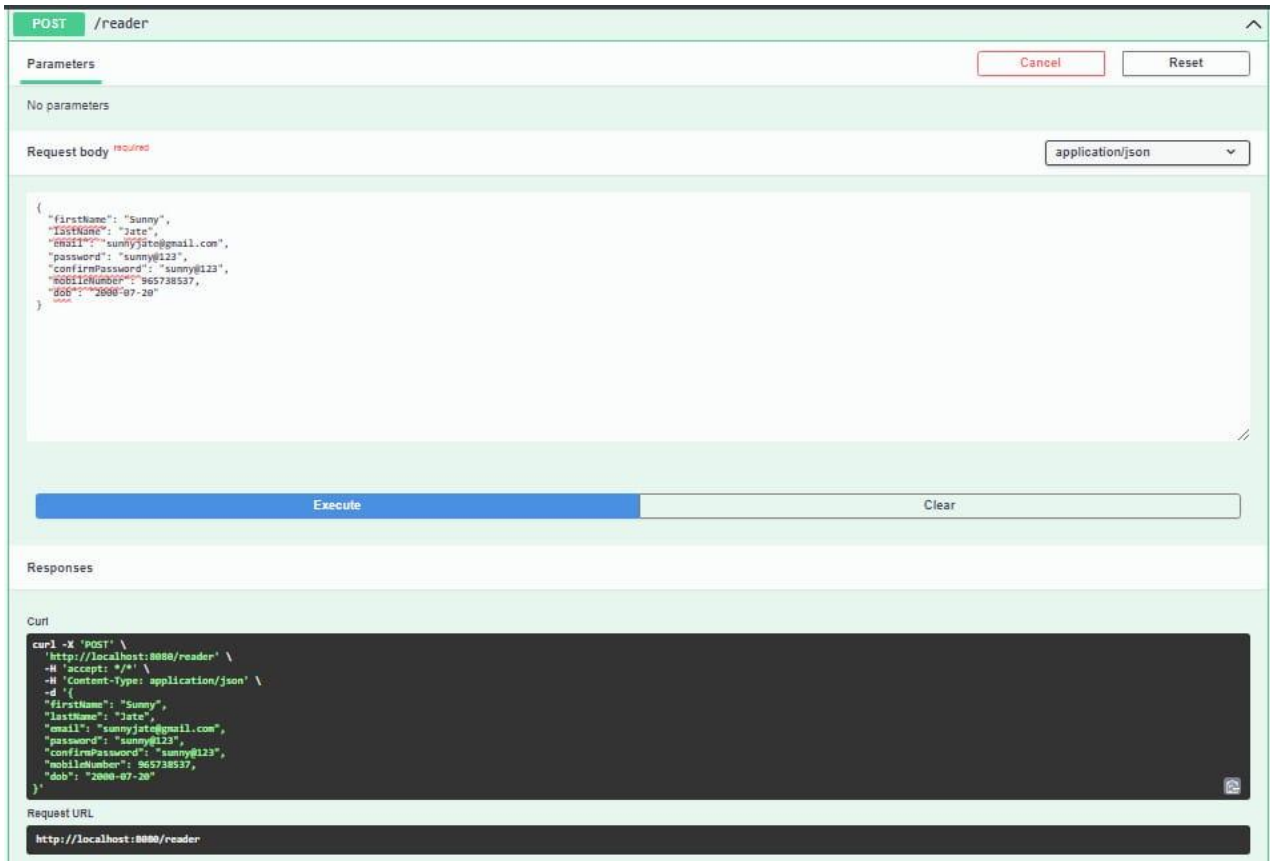
Mobile Number

Date of Birth

mm/dd/yyyy

Already have an account? Signin here

## Reader Controller



No parameters

Request body required

```
{
  "firstName": "Sunny",
  "lastName": "Jate",
  "email": "sunnyjate@gmail.com",
  "password": "sunny@123",
  "confirmPassword": "sunny@123",
  "mobileNumber": 965738537,
  "dob": "2000-07-20"
}
```

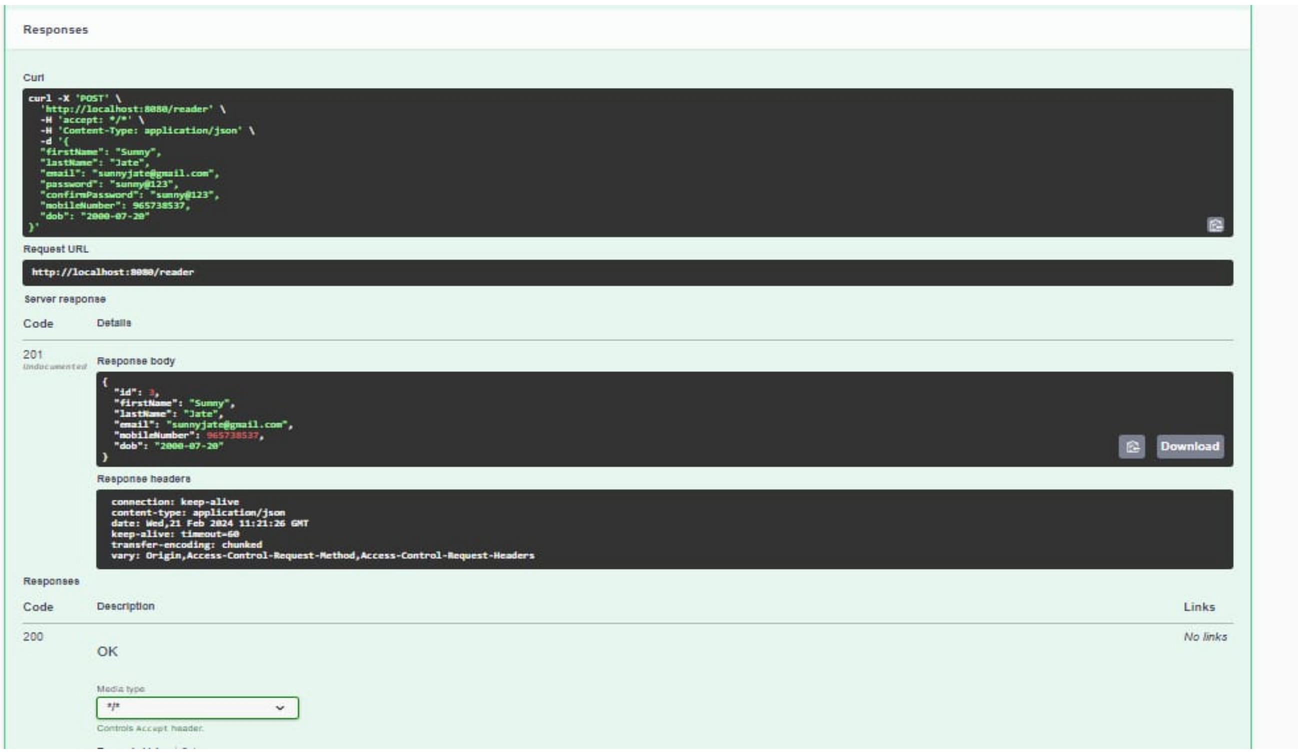
Responses

Curl

```
curl -X 'POST' \
  'http://localhost:8080/reader' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{
  "firstName": "Sunny",
  "lastName": "Jate",
  "email": "sunnyjate@gmail.com",
  "password": "sunny@123",
  "confirmPassword": "sunny@123",
  "mobileNumber": 965738537,
  "dob": "2000-07-20"
}'
```

Request URL

```
http://localhost:8080/reader
```



Responses

Curl

```
curl -X 'POST' \
  'http://localhost:8080/reader' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{
  "firstName": "Sunny",
  "lastName": "Jate",
  "email": "sunnyjate@gmail.com",
  "password": "sunny@123",
  "confirmPassword": "sunny@123",
  "mobileNumber": 965738537,
  "dob": "2000-07-20"
}'
```

Request URL

```
http://localhost:8080/reader
```

Server response

Code Details

201 *Undocumented*

Response body

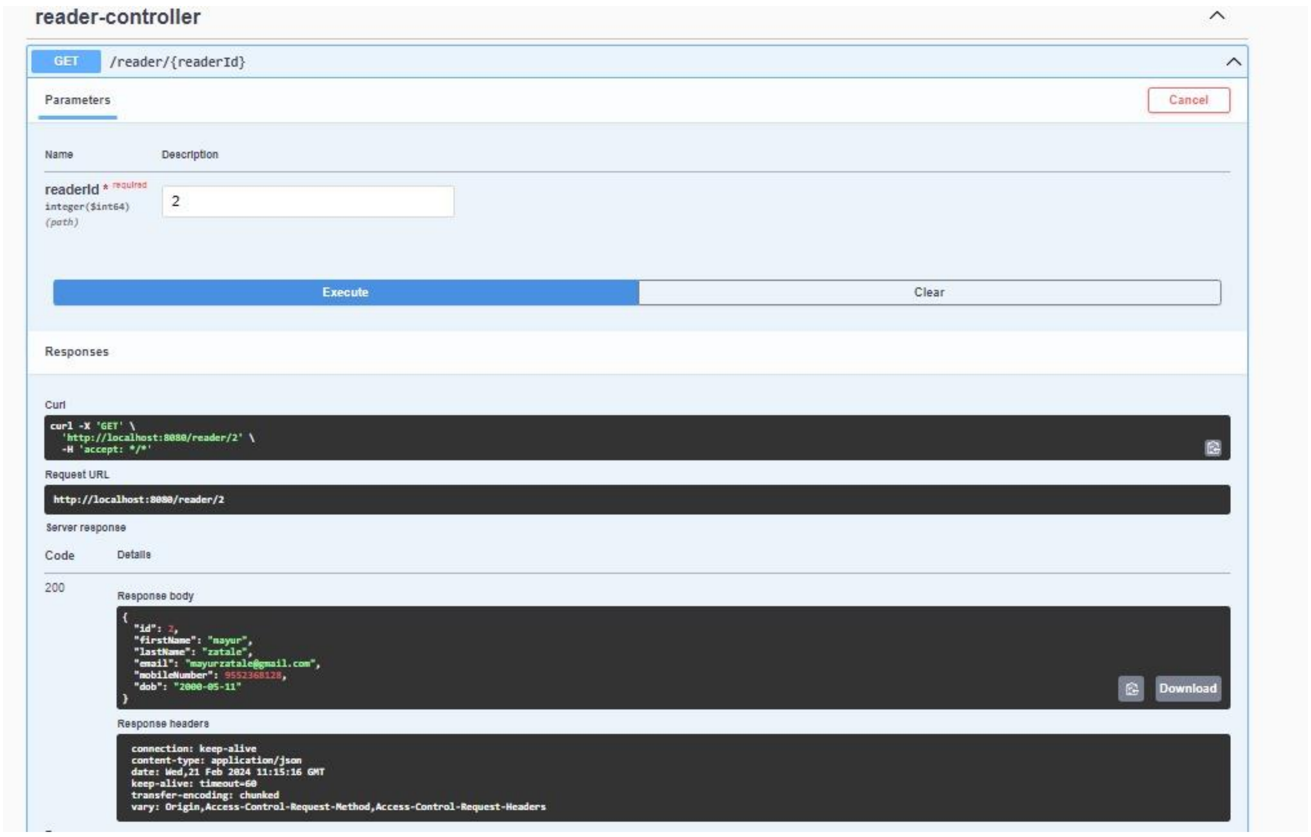
```
{
  "id": 3,
  "firstName": "Sunny",
  "lastName": "Jate",
  "email": "sunnyjate@gmail.com",
  "mobileNumber": 965738537,
  "dob": "2000-07-20"
}
```

Response headers

```
connection: keep-alive
content-type: application/json
date: Wed,21 Feb 2024 11:21:26 GMT
keep-alive: timeout=60
transfer-encoding: chunked
vary: Origin,Access-Control-Request-Method,Access-Control-Request-Headers
```

Responses

| Code | Description | Links |
|---|---|---|
| 200 | OK<br>Media type<br>*/*<br>Controls Accept header. | *No links* |


reader-controller

GET /reader/{readerId}

Parameters

Name Description

readerId * required
integer($int64)
(path)

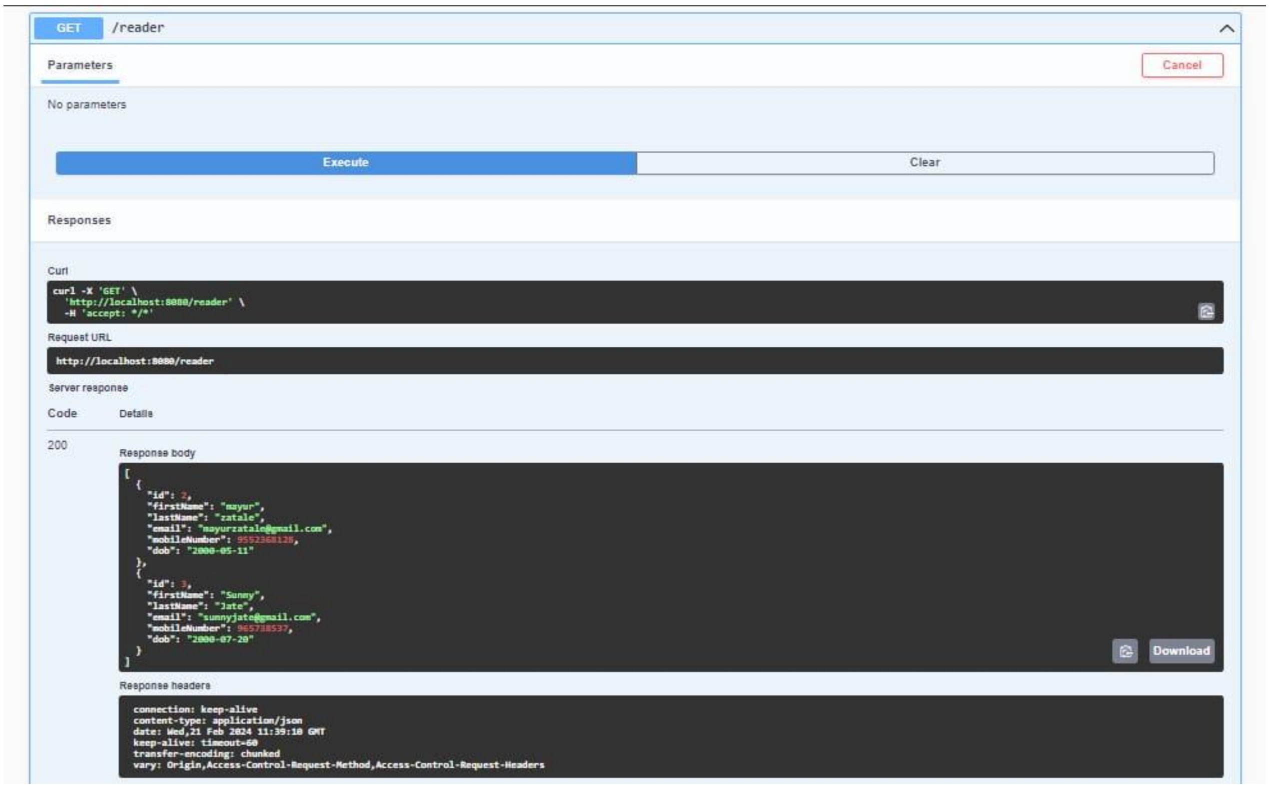
2

Curl
```
curl -X 'GET' \
  'http://localhost:8080/reader/2' \
  -H 'accept: */*'
```
Request URL
```
http://localhost:8080/reader/2
```
Server response

Code Details

200

Response body
```
{
  "id": 2,
  "firstName": "mayur",
  "lastName": "zatale",
  "email": "mayurzatale@gmail.com",
  "mobileNumber": 9552368128,
  "dob": "2000-05-11"
}
```
Response headers
```
 connection: keep-alive
 content-type: application/json
 date: Wed,21 Feb 2024 11:15:16 GMT
 keep-alive: timeout=60
 transfer-encoding: chunked
 vary: Origin,Access-Control-Request-Method,Access-Control-Request-Headers
```


GET /reader

Parameters

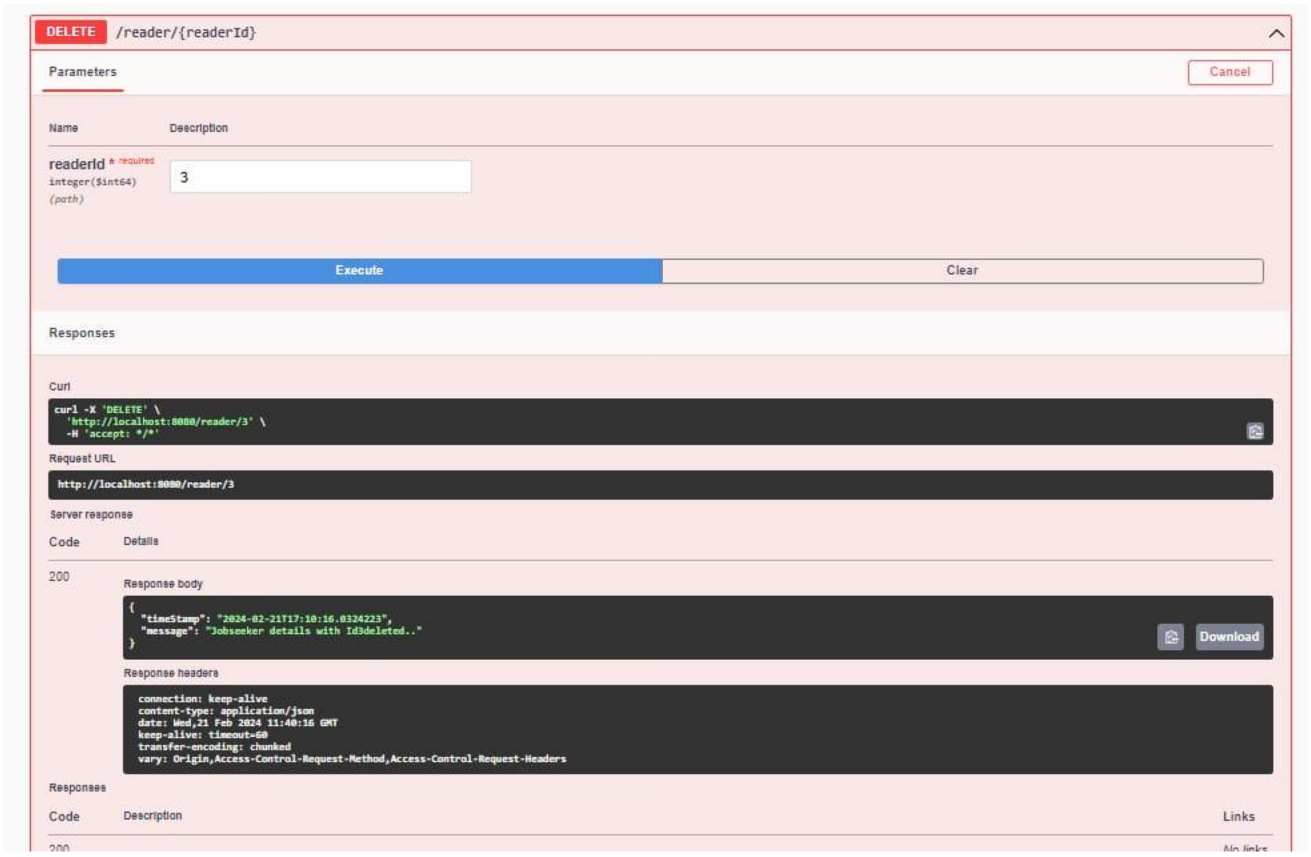
No parameters

Curl
```
curl -X 'GET' \
  'http://localhost:8080/reader' \
  -H 'accept: */*'
```
Request URL
```
http://localhost:8080/reader
```
Server response

Code Details

200

Response body
```
[
  {
    "id": 2,
    "firstName": "mayur",
    "lastName": "zatale",
    "email": "mayurzatale@gmail.com",
    "mobileNumber": 9552368128,
    "dob": "2000-05-11"
  },
  {
    "id": 3,
    "firstName": "Sunny",
    "lastName": "Jate",
    "email": "sunnyjate@gmail.com",
    "mobileNumber": 965738537,
    "dob": "2000-07-20"
  }
]
```
Response headers
```
 connection: keep-alive
 content-type: application/json
 date: Wed,21 Feb 2024 11:39:10 GMT
 keep-alive: timeout=60
 transfer-encoding: chunked
 vary: Origin,Access-Control-Request-Method,Access-Control-Request-Headers
```



Curl

```
curl -X 'DELETE' \
  'http://localhost:8080/reader/3' \
  -H 'accept: */*'
```

Request URL

```
http://localhost:8080/reader/3
```

Server response

Code Details

200

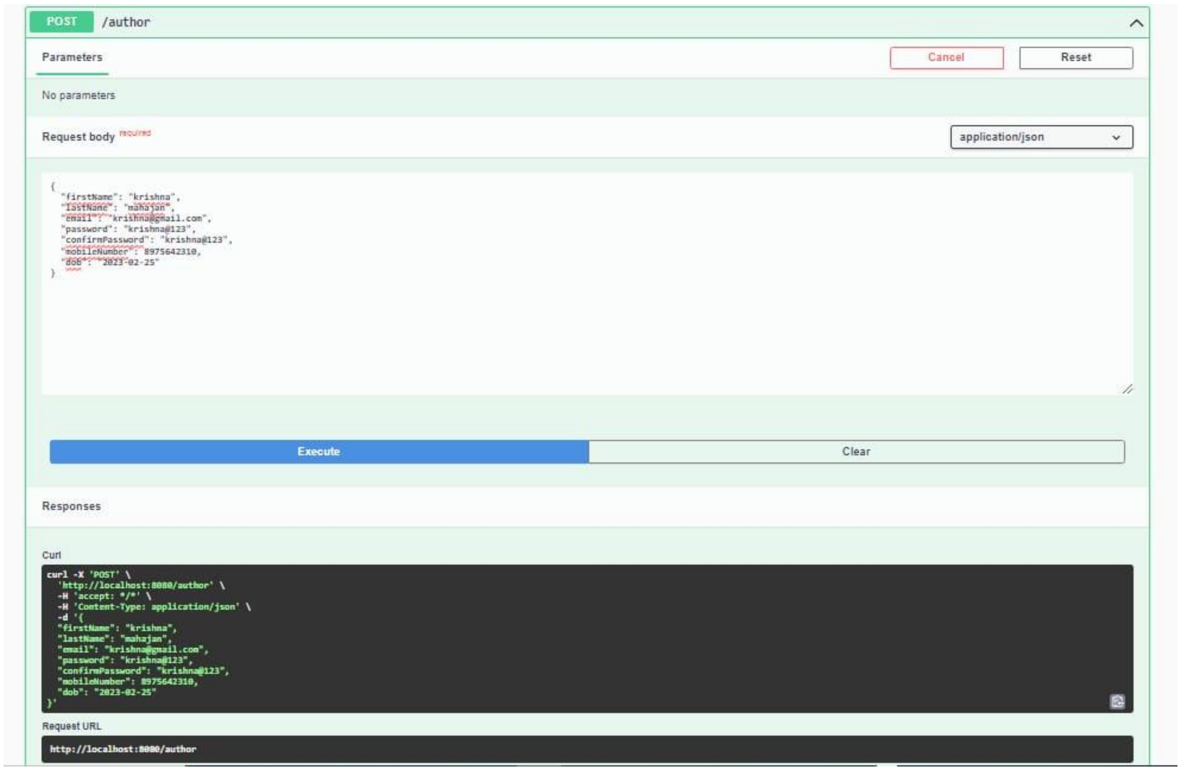
Response body

```
{
  "timeStamp": "2024-02-21T17:10:16.0324223",
  "message": "Jobseeker details with Id3deleted.."
}
```

Response headers

```
 connection: keep-alive
 content-type: application/json
 date: Wed,21 Feb 2024 11:40:16 GMT
 keep-alive: timeout=60
 transfer-encoding: chunked
 vary: Origin,Access-Control-Request-Method,Access-Control-Request-Headers
```

## Author Controller



No parameters

Request body required

```
{
  "firstName": "krishna",
  "lastName": "mahajan",
  "email": "krishna@gmail.com",
  "password": "krishna@123",
  "confirmPassword": "krishna@123",
  "mobileNumber": 8975642310,
  "dob": "2023-02-25"
}
```

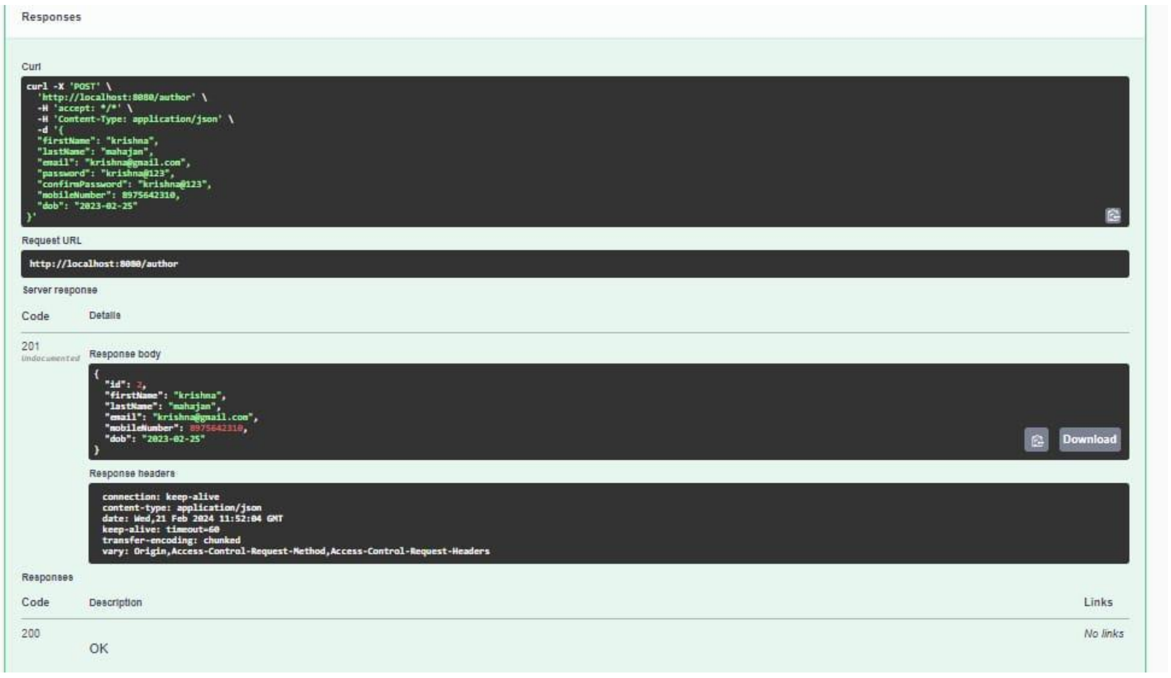
Responses

Curl

```
curl -X 'POST' \
  'http://localhost:8080/author' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{
  "firstName": "krishna",
  "lastName": "mahajan",
  "email": "krishna@gmail.com",
  "password": "krishna@123",
  "confirmPassword": "krishna@123",
  "mobileNumber": 8975642310,
  "dob": "2023-02-25"
}'
```

Request URL

```
http://localhost:8080/author
```



Responses

Curl

```
curl -X 'POST' \
  'http://localhost:8080/author' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{
  "firstName": "krishna",
  "lastName": "mahajan",
  "email": "krishna@gmail.com",
  "password": "krishna@123",
  "confirmPassword": "krishna@123",
  "mobileNumber": 8975642310,
  "dob": "2023-02-25"
}'
```

Request URL

```
http://localhost:8080/author
```

Server response

| Code | Details |
|---|---|
| 201 *Undocumented* | Response body |

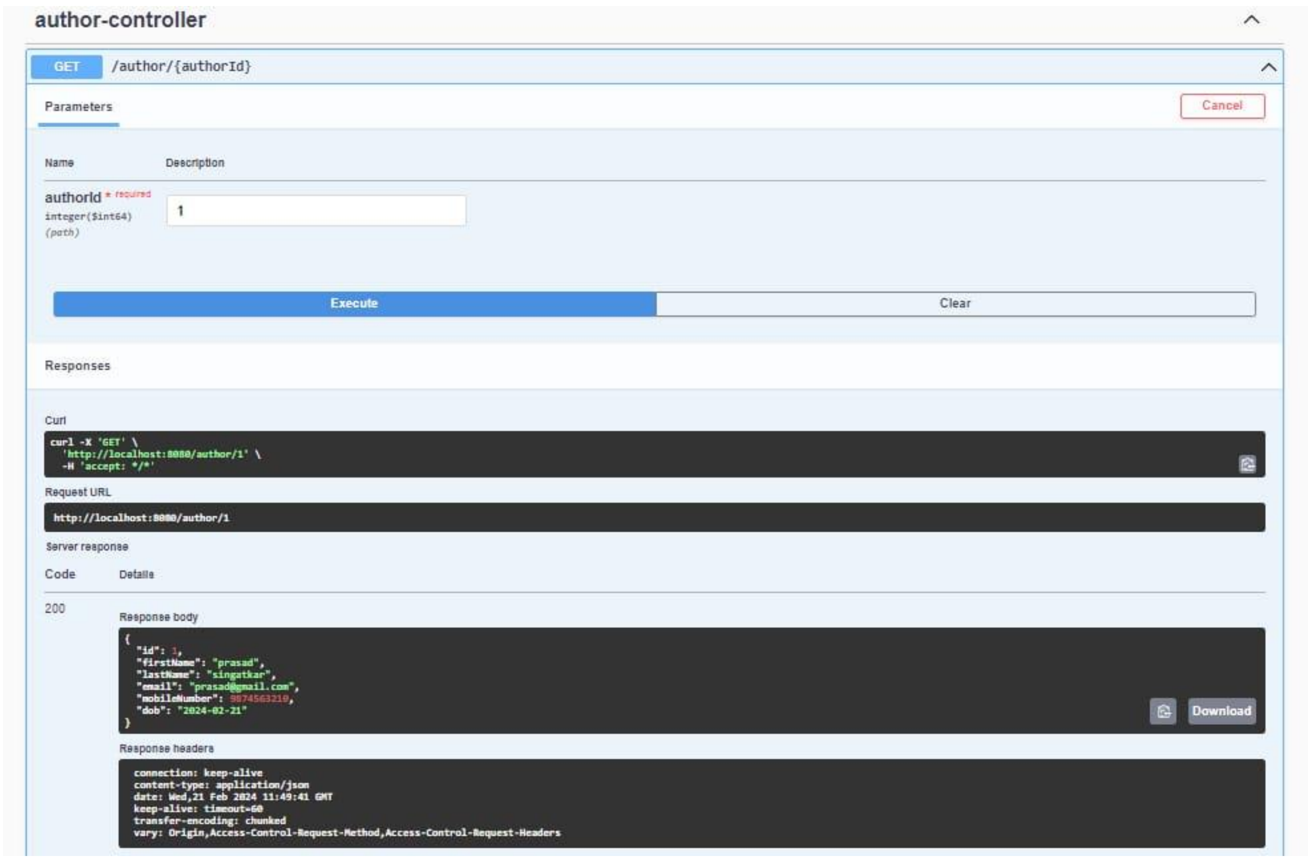
```
{
  "id": 2,
  "firstName": "krishna",
  "lastName": "mahajan",
  "email": "krishna@gmail.com",
  "mobileNumber": 8975642310,
  "dob": "2023-02-25"
}
```

Response headers

```
 connection: keep-alive
 content-type: application/json
 date: Wed,21 Feb 2024 11:52:04 GMT
 keep-alive: timeout=60
 transfer-encoding: chunked
 vary: Origin,Access-Control-Request-Method,Access-Control-Request-Headers
```

Responses

| Code | Description | Links |
|---|---|---|
| 200 | OK | *No links* |



/author/{authorId}

| Name | Description |
|---|---|
| authorId * required<br>integer($int64)<br>(path) | 1 |

Responses

Curl

```
curl -X 'GET' \
  'http://localhost:8080/author/1' \
  -H 'accept: */*'
```

Request URL

```
http://localhost:8080/author/1
```

Server response

Code Details

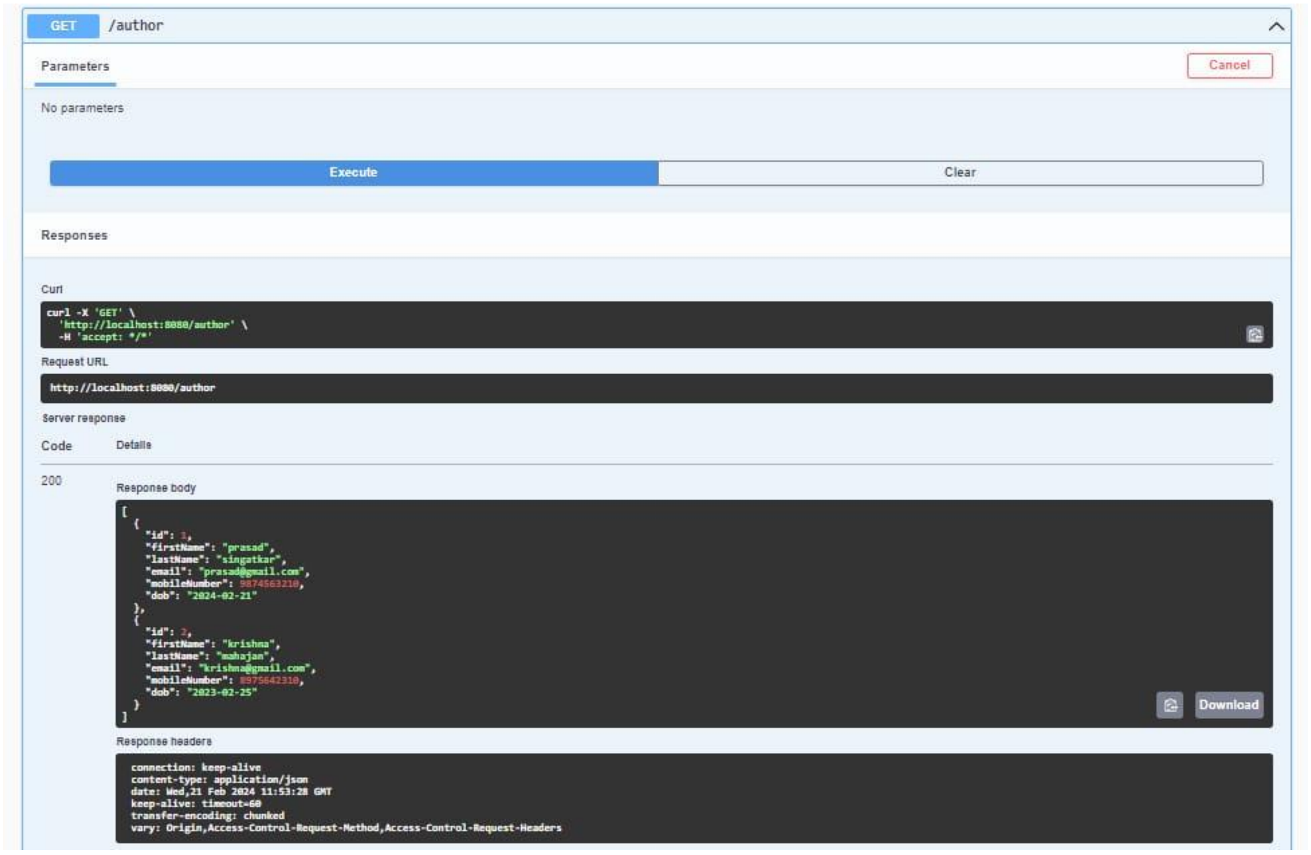
200

Response body

```
{
  "id": 1,
  "firstName": "prasad",
  "lastName": "singatkar",
  "email": "prasad@gmail.com",
  "mobileNumber": 9874563210,
  "dob": "2024-02-21"
}
```

Response headers

```
 connection: keep-alive
 content-type: application/json
 date: Wed,21 Feb 2024 11:49:41 GMT
 keep-alive: timeout=60
 transfer-encoding: chunked
 vary: Origin,Access-Control-Request-Method,Access-Control-Request-Headers
```



/author

No parameters

Responses

Curl

```
curl -X 'GET' \
  'http://localhost:8080/author' \
  -H 'accept: */*'
```

Request URL

```
http://localhost:8080/author
```

Server response

Code Details

200

Response body

```
[
  {
    "id": 1,
    "firstName": "prasad",
    "lastName": "singatkar",
    "email": "prasad@gmail.com",
    "mobileNumber": 9874563210,
    "dob": "2024-02-21"
  },
  {
    "id": 2,
    "firstName": "krishna",
    "lastName": "mahajan",
    "email": "krishna@gmail.com",
    "mobileNumber": 8975642310,
    "dob": "2023-02-25"
  }
]
```

Response headers

```
 connection: keep-alive
 content-type: application/json
 date: Wed,21 Feb 2024 11:53:28 GMT
 keep-alive: timeout=60
 transfer-encoding: chunked
 vary: Origin,Access-Control-Request-Method,Access-Control-Request-Headers
```

## REFERENCES

http://www.javatpoint.com/java- tutorial
http://www.w3.org
http://www.wikipedia.org
https://www.tutorialspoint.com/java