

## 1. MicroServices to fetch the API and store the data in the Database, Publish the User data to the frontend and update the data in the table

Fetches API data using **AXIOS** and stores it in **MongoDB** database.

### Model Script

```
const mongoose = require("mongoose");

const DataSchema = mongoose.Schema(
  {
    name: {
      type: String,
      require: true,
    },
    email: {
      type: String,
      require: true,
    },
    gender: {
      type: String,
      require: true,
    },
    status: {
      type: String,
      require: true,
    },
  },
  {
    timestamps: true,
  }
);

const Data = mongoose.model("Data", DataSchema);
module.exports = Data;
```

## Fetches Data and Stores in MongoDB

```
try {
  return await axios
    .get("https://gorest.co.in/public-api/users", {
      headers: { "Accept-Encoding": "gzip,deflate,compress" },
    })
    .then((res) => {
      console.log(res.data);
      res.data.data.forEach(async (ele) => {
        const data = new Data({
          id: ele.id,
          name: ele.name,
          email: ele.email,
          gender: ele.gender,
          status: ele.status,
        });
        await data.save();
      });
    });
} catch (error) {
  console.error(error);
}
```

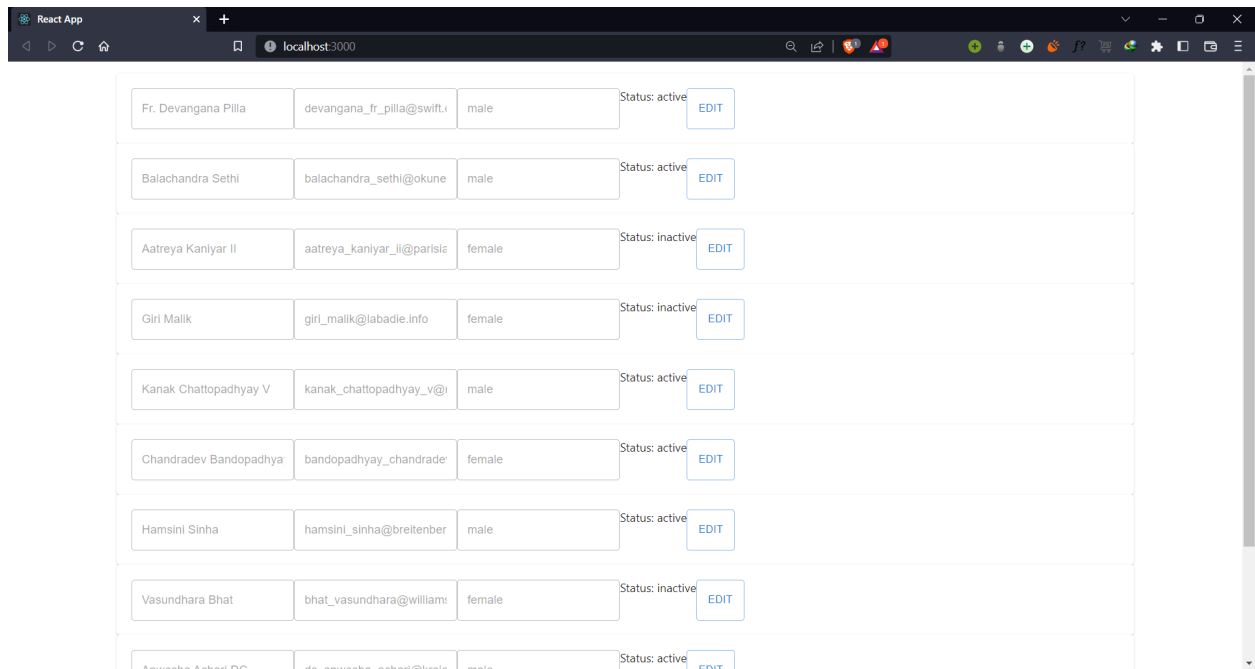
## MongoDB Data

QUERY RESULTS: 1-10 OF 10

```
_id: ObjectId('63a2df635d93f6d21ed823ea')
name: "Fr. Devangana Pilla"
email: "devangana_fr_pilla@swift.co"
gender: "male"
status: "active"
createdAt: 2022-12-21T10:26:43.681+00:00
updatedAt: 2022-12-21T12:07:11.316+00:00
__v: 0
```

## 2. FrontEnd development to view and update the user data

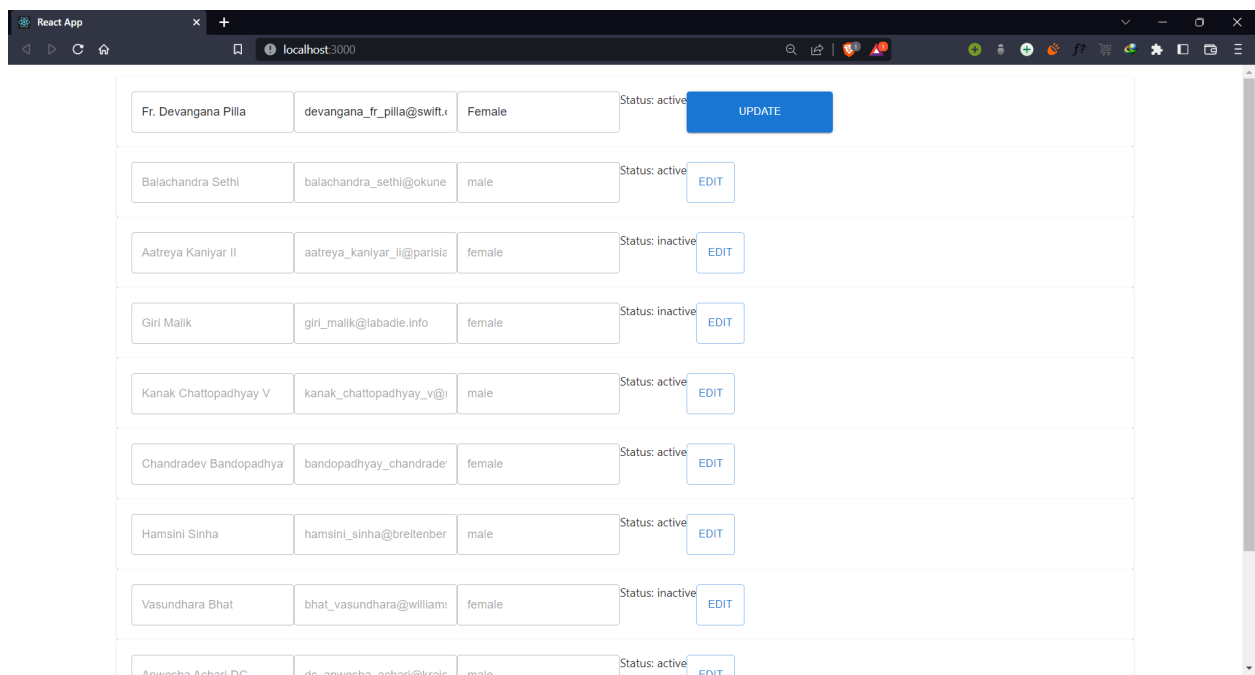
### Published the data in frontend



The screenshot shows a web application running on localhost:3000. It displays a table with user information. Each row contains the user's name, email, gender, status, and an 'EDIT' button.

Fr. Devangana Pilla	devangana_fr_pilla@swift.i	male	Status: active	EDIT
Balachandra Sethi	balachandra_sethi@okune	male	Status: active	EDIT
Aatreya Kaniyar II	aatreya_kaniyar_ii@parisia	female	Status: inactive	EDIT
Giri Malik	giri_malik@labadie.info	female	Status: inactive	EDIT
Kanak Chattopadhyay V	kanak_chattopadhyay_v@i	male	Status: active	EDIT
Chandradev Bandopadhyay	bandopadhyay_chandradev	female	Status: active	EDIT
Hamsini Sinha	hamsini_sinha@breitenber	male	Status: active	EDIT
Vasundhara Bhat	bhat_vasundhara@william	female	Status: inactive	EDIT
Anuscha Ashori DC	dc_anuscha_ashori@krate	male	Status: active	EDIT

### Edit Fields



The screenshot shows the same web application, but the first row now has an 'UPDATE' button instead of an 'EDIT' button, indicating that the edit functionality is active for that user.

Fr. Devangana Pilla	devangana_fr_pilla@swift.i	Female	Status: active	UPDATE
Balachandra Sethi	balachandra_sethi@okune	male	Status: active	EDIT
Aatreya Kaniyar II	aatreya_kaniyar_ii@parisia	female	Status: inactive	EDIT
Giri Malik	giri_malik@labadie.info	female	Status: inactive	EDIT
Kanak Chattopadhyay V	kanak_chattopadhyay_v@i	male	Status: active	EDIT
Chandradev Bandopadhyay	bandopadhyay_chandradev	female	Status: active	EDIT
Hamsini Sinha	hamsini_sinha@breitenber	male	Status: active	EDIT
Vasundhara Bhat	bhat_vasundhara@william	female	Status: inactive	EDIT
Anuscha Ashori DC	dc_anuscha_ashori@krate	male	Status: active	EDIT

### 3. Microservice to export the data into CSV

```
try {
  const data = await Data.find();
  const csvFields = [
    "Name",
    "Email",
    "Gender",
    "Status",
    "CreatedAt",
    "UpdatedAt",
  ];
  const csvParser = new CsvParser({ csvFields });
  const csvData = csvParser.parse(data);

  res.setHeader("Content-Type", "text/csv");
  res.setHeader("Content-Disposition", "attachment;
filename=data.csv");

  res.status(200).end(csvData);
  // res.status(200).json({ data: data });
} catch (err) {
  res.status(500).send({ message: err.message ||
"Internal Server Error!" });
}
```

Export CSV from MongoDB and Sending to Frontend using Node.JS

Saved CSV file from frontend

```
const getCSVHandler = () => {
  return axios({
    method: "GET",
    url: "http://localhost:5000/api/getCSV",
    responseType: "arraybuffer",
    headers: {
      Accept: "text/csv",
    },
  })
  .then((response) => {
    console.log(response);
    if (response.status === 200 && response) {
      return response.data;
    }
  })
  .catch((error) => console.log(error));
};

const onHandleDownload = () => {
  getCSVHandler().then((data) => {
    const blob = new Blob([data], {
      type: "text/csv",
    });
    saveAs(blob, "csv_name.csv");
  });
};
```

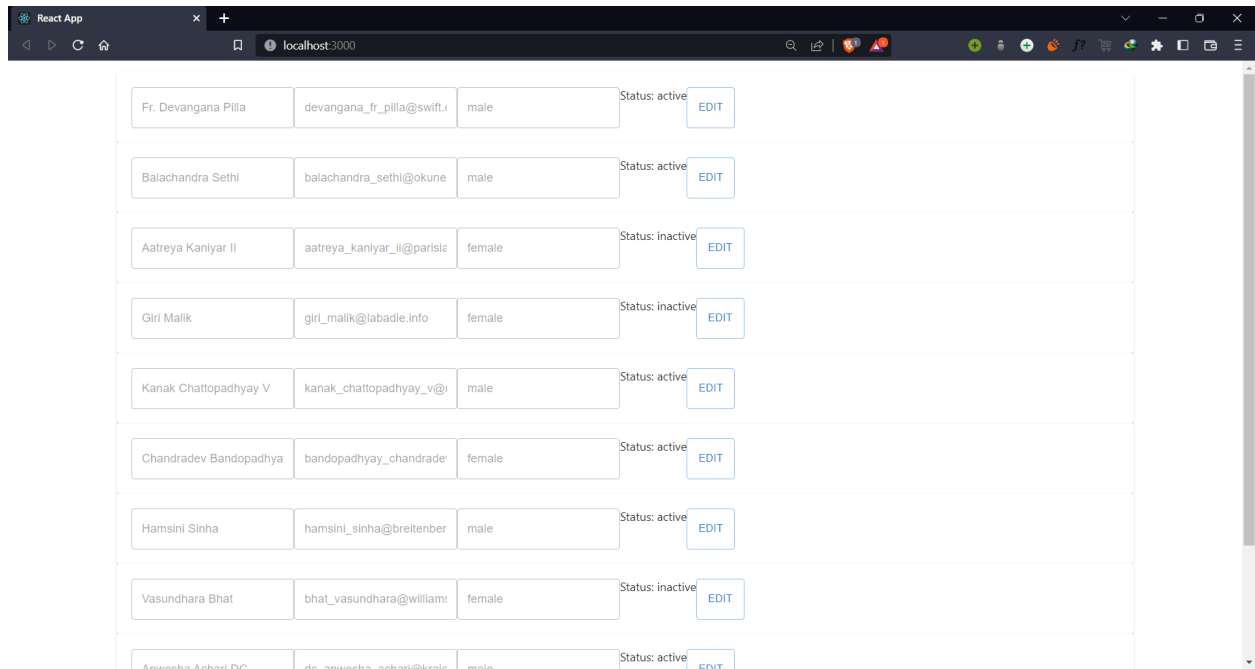
## 5.DB Script

### Fetching Data & Editing Data

```
//fetch data
exports.getData = async (req, res) => {
  console.log("Reached Backend");
  try {
    const data = await Data.find();
    res.status(200).json({ data: data });
  } catch (err) {
    res.status(500).send({ message: err.message || "Internal
Server Error!" });
  }
};

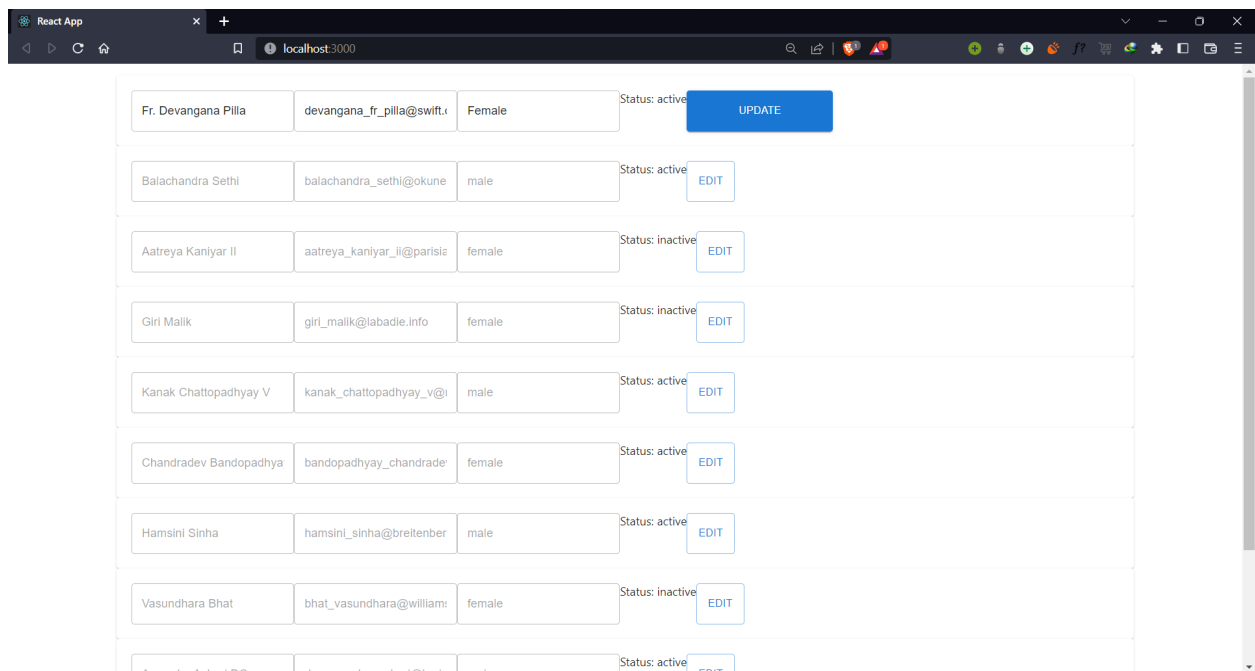
//edit data
exports.editData = async (req, res) => {
  console.log("Reached Backend");
  let body = {
    name: req.body.name,
    email: req.body.email,
    status: req.body.status,
    gender: req.body.gender,
  };
  try {
    await Data.findByIdAndUpdate(req.body._id, {
      ...body,
    });
    res.status(200).json({ message: "Data updated!" });
  } catch (err) {
    console.log(err);
    res.status(500).send({ message: err.message || "Internal
Server Error!" });
  }
};
```

## 6. Frontend webpage screenshots



This screenshot shows a web application running on localhost:3000. It displays a list of users in a table. Each row contains the user's name, email, gender, status, and an 'EDIT' button. The data is as follows:

Name	Email	Gender	Status	Action
Fr. Devangana Pilla	devangana_fr_pilla@swift.i	male	Status: active	EDIT
Balachandra Sethi	balachandra_sethi@okune	male	Status: active	EDIT
Aatreya Kaniyar II	aatreya_kaniyar_ii@parisia	female	Status: inactive	EDIT
Giri Malik	giri_malik@labadie.info	female	Status: inactive	EDIT
Kanak Chattopadhyay V	kanak_chattopadhyay_v@i	male	Status: active	EDIT
Chandradev Bandopadhyay	bandopadhyay_chandradev	female	Status: active	EDIT
Hamsini Sinha	hamsini_sinha@breitenber	male	Status: active	EDIT
Vasundhara Bhat	bhat_vasundhara@william	female	Status: inactive	EDIT
Anuscha Ashraf DC	dc_anuscha_ashraf@krala	male	Status: active	EDIT

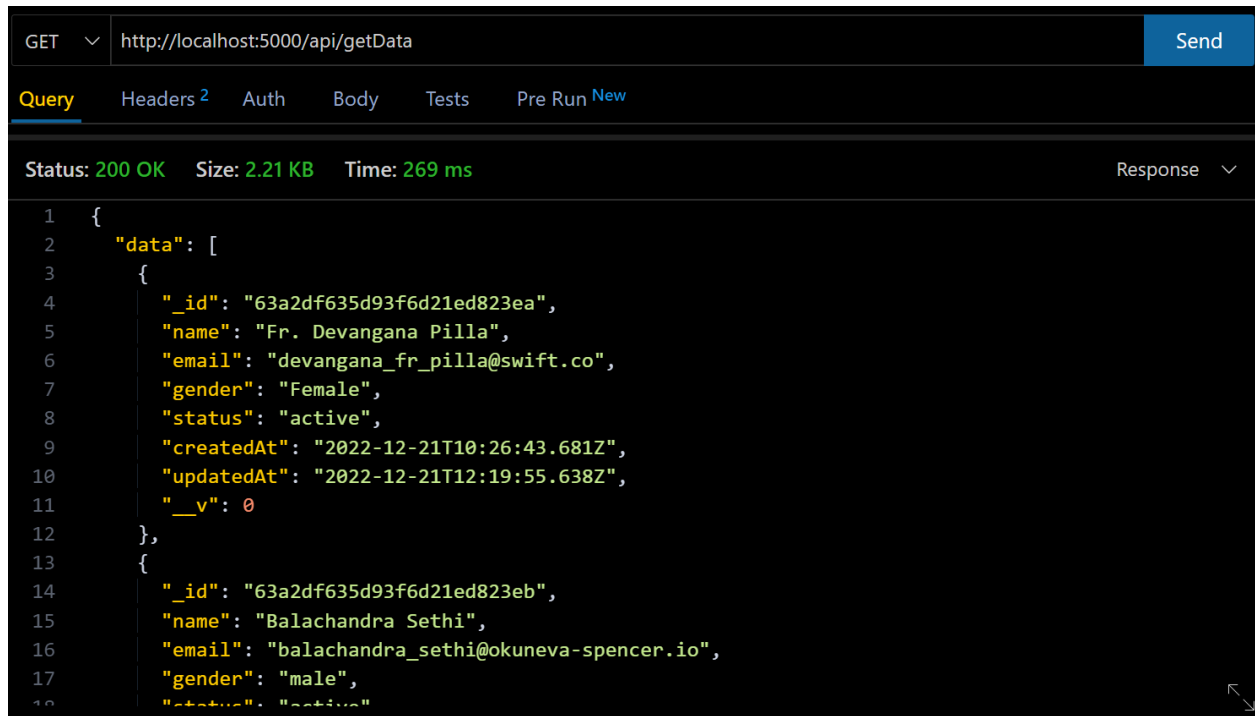


This screenshot shows the same web application, but with a modification: the first row now has an 'UPDATE' button instead of an 'EDIT' button. The data is as follows:

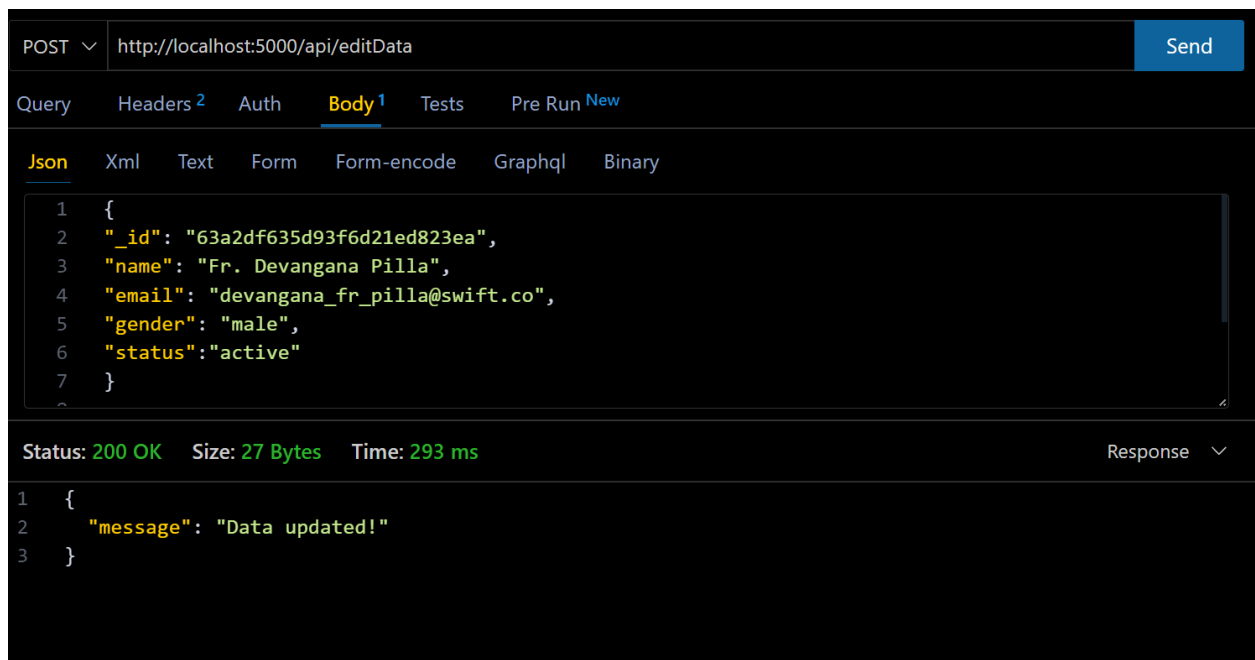
Name	Email	Gender	Status	Action
Fr. Devangana Pilla	devangana_fr_pilla@swift.i	Female	Status: active	UPDATE
Balachandra Sethi	balachandra_sethi@okune	male	Status: active	EDIT
Aatreya Kaniyar II	aatreya_kaniyar_ii@parisia	female	Status: inactive	EDIT
Giri Malik	giri_malik@labadie.info	female	Status: inactive	EDIT
Kanak Chattopadhyay V	kanak_chattopadhyay_v@i	male	Status: active	EDIT
Chandradev Bandopadhyay	bandopadhyay_chandradev	female	Status: active	EDIT
Hamsini Sinha	hamsini_sinha@breitenber	male	Status: active	EDIT
Vasundhara Bhat	bhat_vasundhara@william	female	Status: inactive	EDIT
Anuscha Ashraf DC	dc_anuscha_ashraf@krala	male	Status: active	EDIT

## 7.API access postman screenshots and Postman collection file

### Get Data **GET** request



### Edit Data **POST** request with JSON data





## Requesting CSV file from API

GET

http://localhost:5000/api/getCSV

Send

Query

Headers 2

Auth

Body

Tests

Pre Run New

Query Parameters

☐

parameter

value

Status: 200 OK   Size: 6.07 KB   Time: 249 ms   Response

```
1  "$__", "$isNew", "_doc"
2  "{
  \"activePaths\": {
    \"paths\": {
      \"_id\": \"init\", \"name\": \"init\", \"email\": \"init\", \"gender\": \"init\", \"status\": \"init\", \"createdAt\": \"init\", \"updatedAt\": \"init\", \"__v\": \"init\"
    },
    \"states\": {
      \"init\": {
        \"_id\": true, \"name\": true, \"email\": true, \"gender\": true, \"status\": true, \"createdAt\": true, \"updatedAt\": true, \"__v\": true
      }
    },
    \"skipId\": true, \"false\", {
      \"_id\": \"63a2df635d93f6d21ed823ea\", \"name\": \"Fr. Devangana Pilla\", \"email\": \"devangana_fr_pilla@swift.co\", \"gender\": \"male\", \"status\": \"active\", \"createdAt\": \"2022-12-21T10:26:43.681Z\", \"updatedAt\": \"2022-12-21T12:35:51.294Z\", \"__v\": 0
    }
  }
}
3  \"{
  \"activePaths\": {
    \"paths\": {
      \"_id\": \"init\", \"name\": \"init\", \"email\": \"init\", \"gender\": \"init\", \"status\": \"init\", \"createdAt\": \"init\", \"updatedAt\": \"init\", \"__v\": \"init\"
    },
    \"states\": {
      \"init\": {
        \"_id\": true, \"name\": true, \"email\": true, \"gender\": true, \"status\": true, \"createdAt\": true, \"updatedAt\": true, \"__v\": true
      }
    },
    \"skipId\": true, \"false\", {
      \"_id\": \"63a2df635d93f6d21ed823eb\", \"name\": \"Balachandra Sethi\", \"email\": \"\"
    }
  }
}
```

