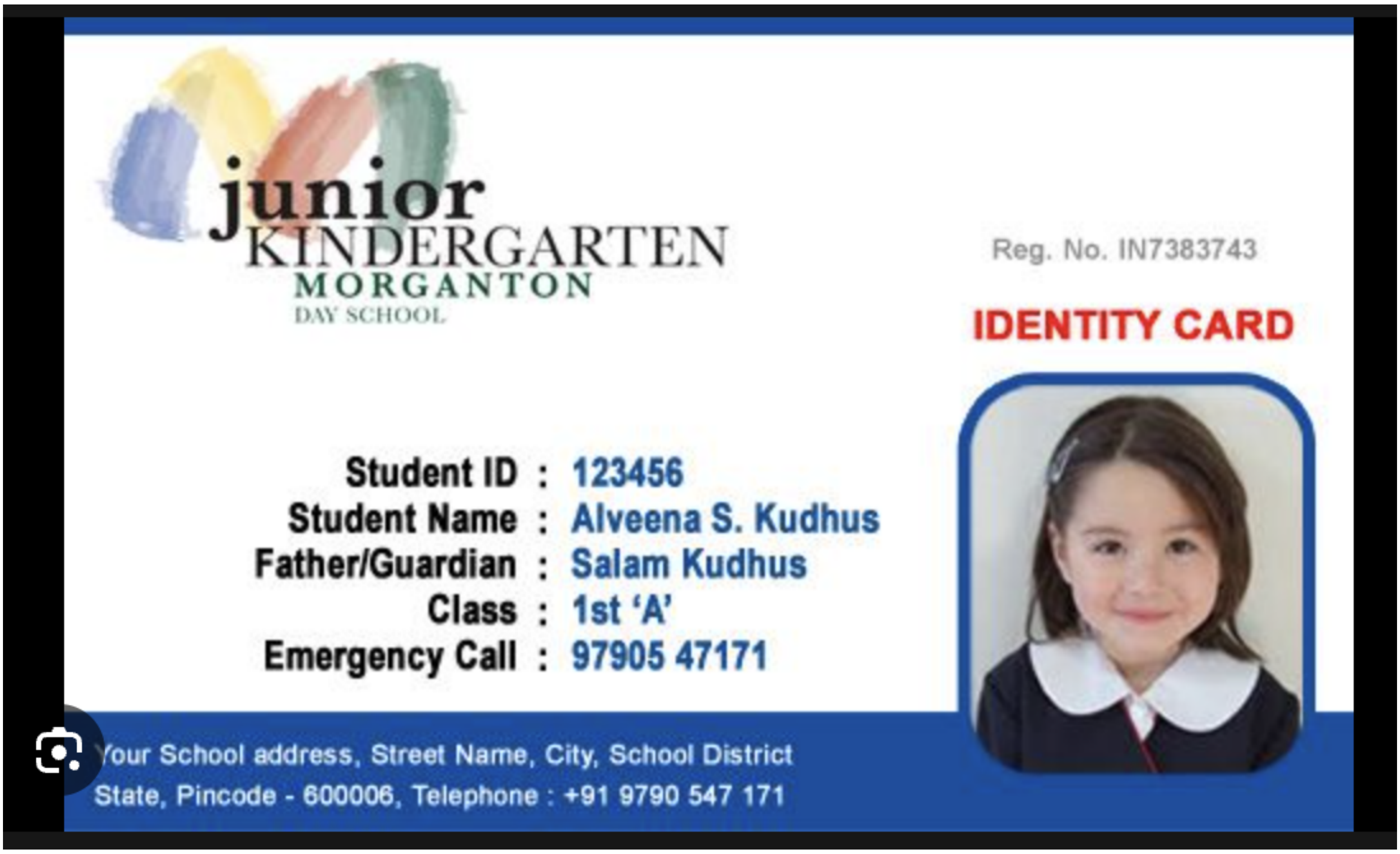
Backend 1.2_CW Exercises

1. learning data modeling

- Create mongoose models for the given images.
- Add data to the model

1.1



junior
KINDERGARTEN
MORGANTON
DAY SCHOOL

Reg. No. IN7383743

IDENTITY CARD

Student ID : 123456
Student Name : Alveena S. Kudhus
Father/Guardian : Salam Kudhus
Class : 1st 'A'
Emergency Call : 97905 47171

Your School address, Street Name, City, School District
State, Pincode - 600006, Telephone : +91 9790 547 171

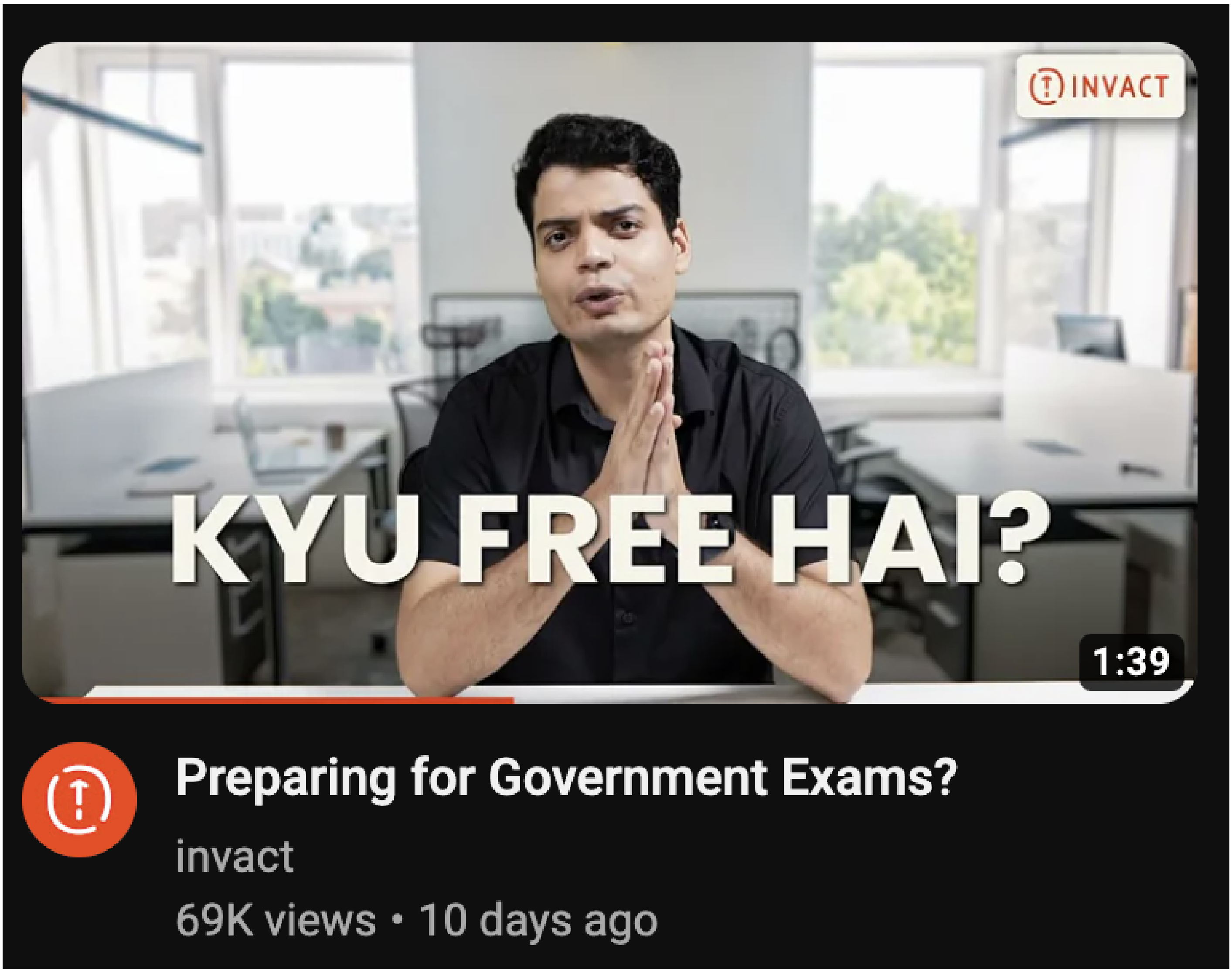
- Solution

```
const mongoose = require('mongoose')

const studentSchema = new mongoose.Schema({
  registrationNumber: String,
  studentId: Number,
  studentName: String,
  studentProfilePicURL: String,
  fatherOrGuardianName: String,
  standard: String,
  emergencyContact: Number,
})

const Student = mongoose.model('Student', studentSchema)
```

```
module.exports = Student

// function to add student data
async function addStudentData() {
  const newStudent = new Student({
    registrationNumber: 'IN7383743',
    studentId: 123456,
    studentName: 'Alveena S. Kudhus',
    fatherGuardianName: 'Salam Kudhus',
    standard: '1st A',
    emergencyContact: 9790547171,
  })

  try {
    const savedStudent = await newStudent.save()
    console.log('Student data saved successfully:', savedStudent)
  } catch (error) {
    console.error('Error saving student data:', error)
  }
}

addStudentData()
```

COPY

1.2



Preparing for Government Exams?

invact

69K views • 10 days ago

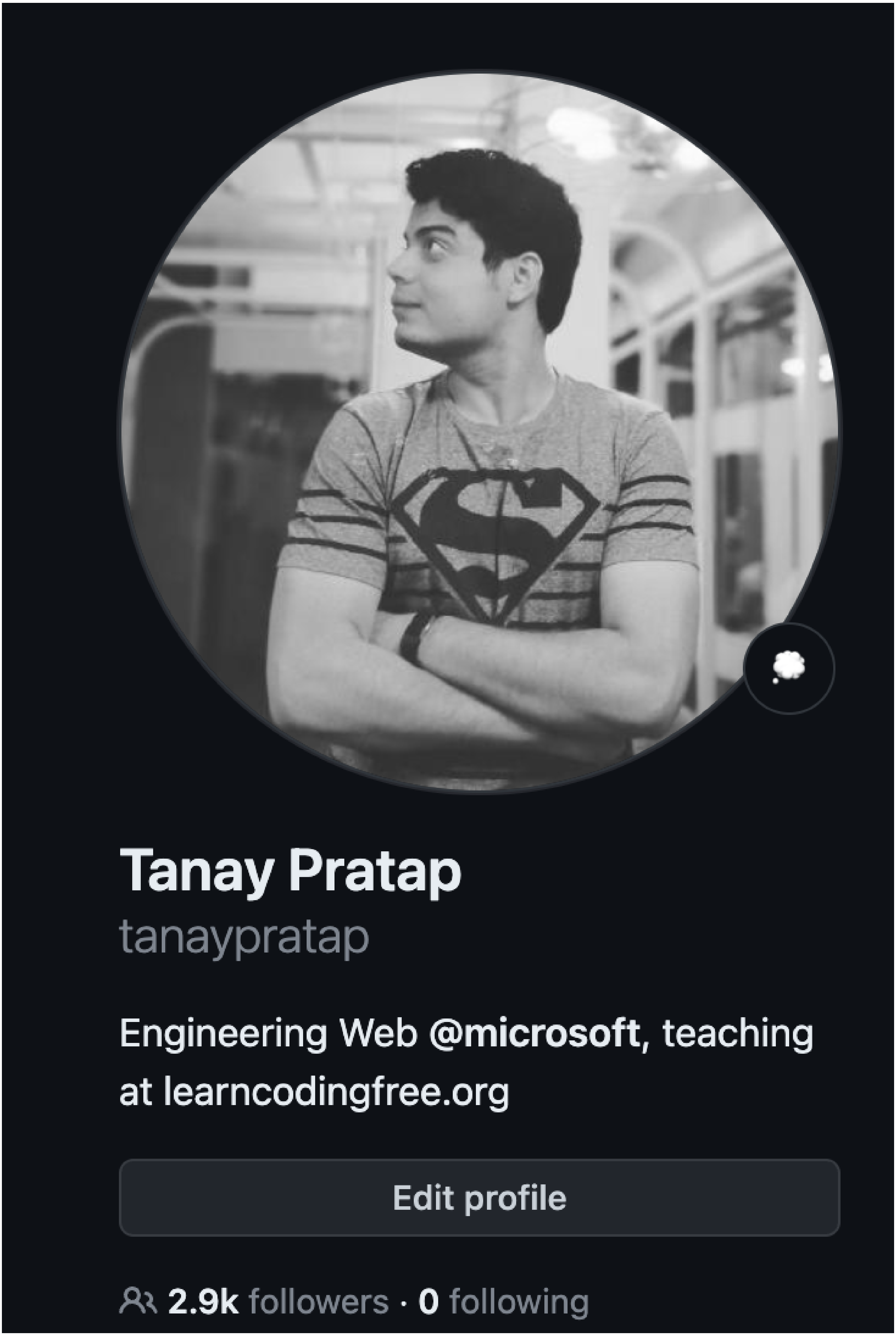
- Solution

  ```
  const mongoose = require('mongoose')

  const videoSchema = new mongoose.Schema({
    videoTitle: String,
    channelName: String,
    channelLogo: String,
    viewsCount: Number,
    thumbnailURl: String,
    totalTime: Number,
    watchedTimeInSeconds: Number,
    postedDate: Date,
    videoURL: String,
  })

  const Video = mongoose.model('Video', videoSchema)

  module.exports = Video

  // function to add video data
  async function addVideoData() {
    const newVideo = new Video({
      videoTitle: 'Preparing for Government Exams?',
      channelName: 'Invact',
      channelLogo: '<imglink>',
      viewsCount: 69000,
      postedDate: '22/08/2023',
  ```

```
  })

  try {
    const savedVideo = await newVideo.save()
    console.log('Video data saved successfully:', savedVideo)
  } catch (error) {
    console.error('Error saving video data:', error)
  }
}

addVideoData()
```

COPY

1.3

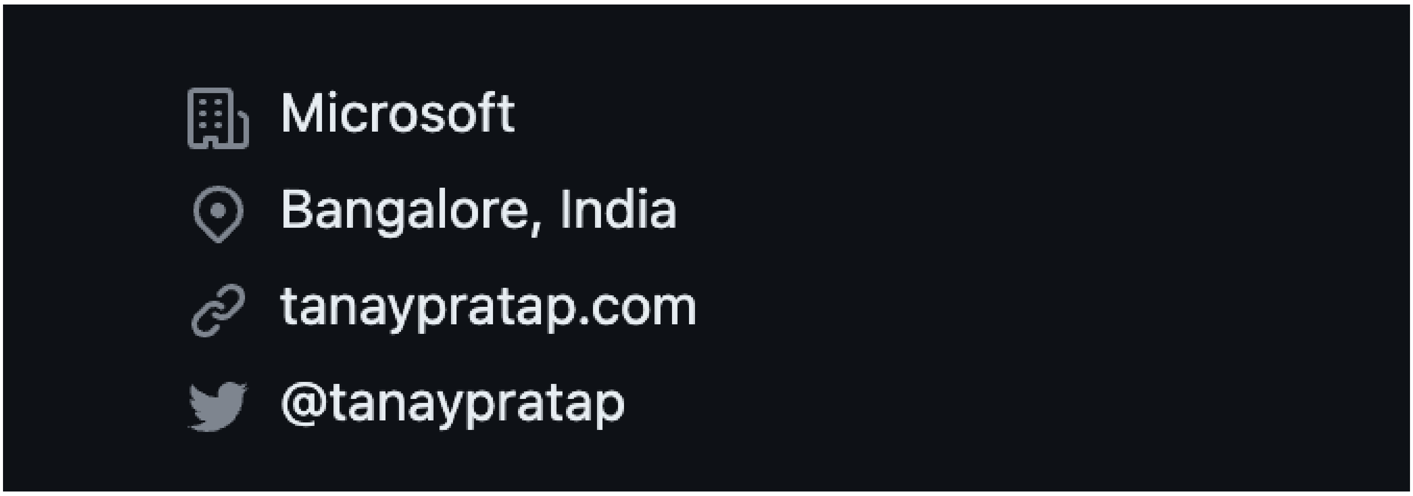


Tanay Pratap

tanaypratap

Engineering Web **@microsoft**, teaching at learncodingfree.org

2.9k followers · **0** following



Microsoft

Bangalore, India

tanaypratap.com

@tanaypratap

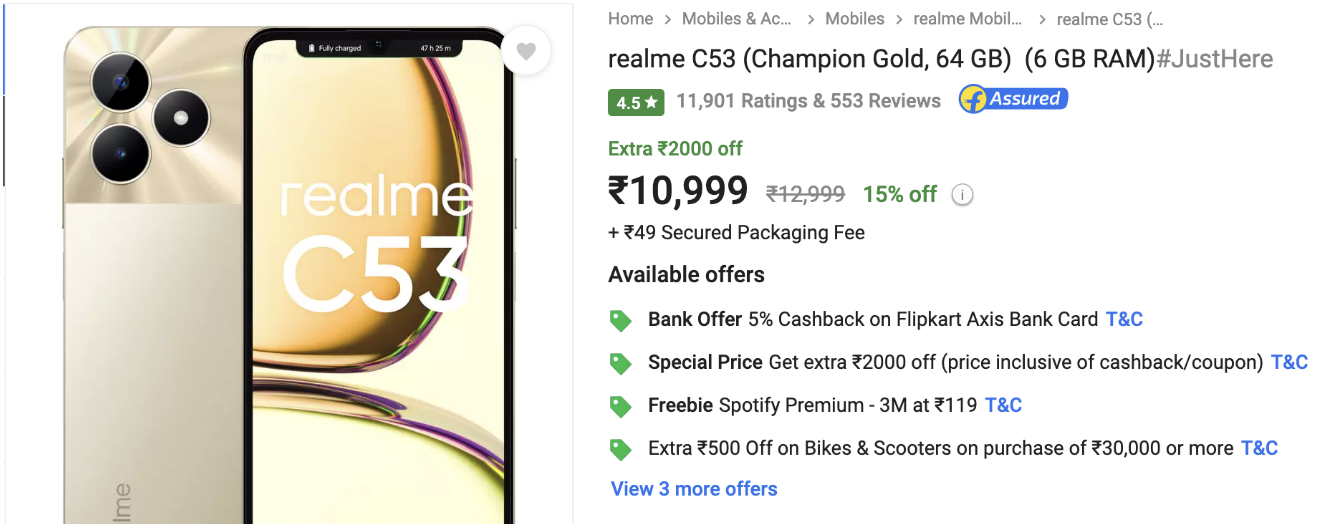
- Solution

```
const mongoose = require('mongoose')

const twitterProfileSchema = new mongoose.Schema({
  fullName: String,
  username: String,
  profilePicURL: String,
  statusURL: String,
  bio: String,
  company: String,
  city: String,
  country: String,
  portfolioURL: String,
  followersCount: Number,
  followingCount: Number,
})

const Profile = mongoose.model('Profile', twitterProfileSchema)

module.exports = Profile

// add profile data
async function addProfileData() {
  const newProfile = new Profile({
    fullName: 'Tanay Pratap',
    username: 'tanaypratap',
    profilePicURL: '<imagelink>',
    bio: 'Engineering Web @microsoft, teaching at learncodingfree.org',
    company: 'Microsoft',
    city: 'Bangalore',
    country: 'India',
    portfolioURL: 'https://tanaypratap.com',
    handle: 'tanaypratap',
    followersCount: 2900,
    followingCount: 0,
  })

  try {
    const savedProfile = await newProfile.save()
    console.log('Profile data saved successfully:', savedProfile)
  } catch (error) {
    console.error('Error saving profile data:', error)
  }
}
```

```
addProfileData()
```

COPY

1.4



Home > Mobiles & Ac... > Mobiles > realme Mobil... > realme C53 (...

realme C53 (Champion Gold, 64 GB) (6 GB RAM)#JustHere

4.5 ★ 11,901 Ratings & 553 Reviews Assured

Extra ₹2000 off

₹10,999 ~~₹12,999~~ 15% off ⓘ

+ ₹49 Secured Packaging Fee

Available offers

- **Bank Offer** 5% Cashback on Flipkart Axis Bank Card T&C
- **Special Price** Get extra ₹2000 off (price inclusive of cashback/coupon) T&C
- **Freebie** Spotify Premium - 3M at ₹119 T&C
- Extra ₹500 Off on Bikes & Scooters on purchase of ₹30,000 or more T&C

View 3 more offers

- Solution

```
{
        category: "Mobile",
        productName: "realme C53",
        productColor: "Champion Gold",
        productImage: "<image source>",
        space: "64 GB",
        ram: "6 GB",
        price: 10999,
        rating: 4.5,
        ratingCount: 11901,
        reviewCount: 553
}
```

COPY

2

Create a model for movie. The model includes the following fields:

- title (String): The title of the movie. This field is required, meaning every movie entry must have a title.
- releaseYear (Number): The year the movie was released. This field is also required.
- genre (Array of Strings): The genre(s) to which the movie belongs. The genre should be one of the predefined values: 'Action', 'Drama', 'Comedy', 'Romance', 'Thriller', 'Fantasy', 'Sci-Fi', 'Horror', 'Sports', 'Musical' or 'Other'.
- director (String): The name of the director of the movie. This field is required.
- actors (Array of Strings): A list of actors' names who were part of the movie's cast.
- language (String): The language in which the movie is presented. This field is required.
- country (String): The country where the movie was produced. The default value is 'India'.

- rating (Number): The movie's rating, represented as a number. The rating must be between 0 and 10. The default value is 0.
- plot (String): A brief description or plot summary of the movie.
- awards (String): Any awards or recognitions the movie has received.
- posterUrl (String): The URL to the poster image of the movie.
- trailerUrl (String): The URL to the official trailer of the movie.
- In the model also include the option `{ timestamps: true }`, which adds `createdAt` and `updatedAt` fields to track the creation and modification times of each movie entry.

Syntax:

```
const movieSchema = new mongoose.Schema({
  title: {
    type: String,
    required: true, // if a field is required
  },
  releaseYear: {
    type: Number,
    required: true,
  },
  genre: [
    {
      type: String,
      enum: [
        'Action',
        'Drama',
        'Comedy',
        'Romance',
        'Thriller',
        'Fantasy',
        'Sci-Fi',
        'Horror',
        'Sports',
        'Musical',
        'Other',
      ],
    },
  ], // syntax for an array of strings
  // enum can be used for pre-defined values
})

const Movie = mongoose.model('Movie', movieSchema)

module.exports = Movie
```

COPY

- Solution https://replit.com/@tanaypratap/BE12CW-2

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

  const movieSchema = new mongoose.Schema({
    title: {
      type: String,
      required: true,
    },
    releaseYear: {
      type: Number,
      required: true,
  ```

```
  },
  genre: [{
    type: String,
    enum: ['Action', 'Drama', 'Comedy', 'Romance', 'Thriller', 'Fantasy', 'Sci-Fi', 'Horr
  }],
  director: {
    type: String,
    required: true,
  },
  actors: [{
    type: String,
  }],
  language: {
    type: String,
    required: true,
  },
  country: {
    type: String,
    default: 'India',
  },
  rating: {
    type: Number,
    min: 0,
    max: 10,
    default: 0,
  },
  plot: {
    type: String,
  },
  awards: {
    type: String,
  },
  posterUrl: {
    type: String,
  },
  trailerUrl: {
    type: String,
  },
 {
  timestamps: true,
});

const Movie = mongoose.model('Movie', movieSchema);

module.exports = Movie;
```

COPY

3

Seeding the data

To update a MongoDB database using Mongoose by seeding it with data from a JSON file, you can follow these general steps:

1. Read JSON File: First create a .json file and put your data in that file. Read the JSON data from your JSON file. You can use built-in Node.js modules like `fs` to accomplish this.
2. Connect to MongoDB: Use Mongoose to connect to your MongoDB database using the `mongoose.connect` method.
3. Define Mongoose Models: Define the Mongoose models for your data, similar to how you've defined the Movie model in your previous examples.
4. Seed Data: Loop through the JSON data and create instances of your Mongoose models using the JSON data.
5. Save Data to Database: For each instance, use the `save` method to save the data to the database.

- JSON file with data

```
;[
  {
    title: 'Dilwale Dulhania Le Jayenge',
    releaseYear: 1995,
    genre: ['Romance', 'Drama'],
    director: 'Aditya Chopra',
    actors: ['Shah Rukh Khan', 'Kajol'],
    language: 'Hindi',
    country: 'India',
    rating: 9.1,
    plot: 'A young man and woman fall in love on a Europe trip.',
    awards: 'Multiple Filmfare Awards',
    posterUrl: 'https://example.com/poster1.jpg',
    trailerUrl: 'https://example.com/trailer1.mp4',
  },
  {
    title: 'Bahubali: The Beginning',
    releaseYear: 2015,
    genre: ['Action', 'Fantasy'],
    director: 'S. S. Rajamouli',
    actors: ['Prabhas', 'Anushka Shetty'],
    language: 'Telugu',
    country: 'India',
    rating: 8.1,
    plot: 'A man embarks on a journey to rescue his mother from a tyrant.',
    awards: 'National Film Award',
    posterUrl: 'https://example.com/poster2.jpg',
    trailerUrl: 'https://example.com/trailer2.mp4',
  },
  {
    title: 'Lagaan',
    releaseYear: 2001,
    genre: ['Drama', 'Sports'],
    director: 'Ashutosh Gowariker',
    actors: ['Aamir Khan', 'Gracy Singh'],
    language: 'Hindi',
    country: 'India',
    rating: 8.2,
    plot: 'A group of villagers challenge British officers to a cricket match.',
    awards: 'Oscar Nomination',
    posterUrl: 'https://example.com/poster3.jpg',
    trailerUrl: 'https://example.com/trailer3.mp4',
  },
  {
    title: 'Kabhi Khushi Kabhie Gham',
    releaseYear: 2001,
    genre: ['Drama', 'Romance'],
    director: 'Karan Johar',
    actors: ['Shah Rukh Khan', 'Kajol'],
    language: 'Hindi',
    country: 'India',
    rating: 7.6,
    plot: 'A family drama spanning generations and continents.',
    awards: 'Multiple Filmfare Awards',
    posterUrl: 'https://example.com/poster4.jpg',
    trailerUrl: 'https://example.com/trailer4.mp4',
  },
  {
    title: 'PK',
    releaseYear: 2014,
    genre: ['Comedy', 'Drama'],
    director: 'Rajkumar Hirani',
    actors: ['Aamir Khan', 'Anushka Sharma'],
    language: 'Hindi',
    country: 'India',
    rating: 8.1,
    plot: 'An alien visits Earth and questions religious beliefs.',
    awards: 'National Film Award',
    posterUrl: 'https://example.com/poster5.jpg',
    trailerUrl: 'https://example.com/trailer5.mp4',
  },
  {
    title: 'Bajrangi Bhaijaan',
    releaseYear: 2015,
    genre: ['Drama', 'Comedy'],
    director: 'Kabir Khan',
    actors: ['Salman Khan', 'Kareena Kapoor'],
    language: 'Hindi',
    country: 'India',
    rating: 8.0,
    plot: 'A man helps a lost girl reunite with her family.',
    awards: 'National Film Award',
    posterUrl: 'https://example.com/poster6.jpg',
    trailerUrl: 'https://example.com/trailer6.mp4',
  },
  {
    title: '3 Idiots',
    releaseYear: 2009,
    genre: ['Comedy', 'Drama'],
    director: 'Rajkumar Hirani',
    actors: ['Aamir Khan', 'Kareena Kapoor'],
    language: 'Hindi',
    country: 'India',
    rating: 8.4,
    plot: 'Two friends search for their long-lost college buddy.',
    awards: 'Multiple Filmfare Awards',
    posterUrl: 'https://example.com/poster7.jpg',
    trailerUrl: 'https://example.com/trailer7.mp4',
  },
  {
    title: 'Gully Boy',
    releaseYear: 2019,
    genre: ['Drama', 'Musical'],
    director: 'Zoya Akhtar',
    actors: ['Ranveer Singh', 'Alia Bhatt'],
    language: 'Hindi',
    country: 'India',
    rating: 7.9,
    plot: 'A young man from the slums aspires to be a rapper.',
    awards: 'Oscar Nomination',
    posterUrl: 'https://example.com/poster8.jpg',
    trailerUrl: 'https://example.com/trailer8.mp4',
  },
]
```

COPY

GitHub Gist with Movie Data

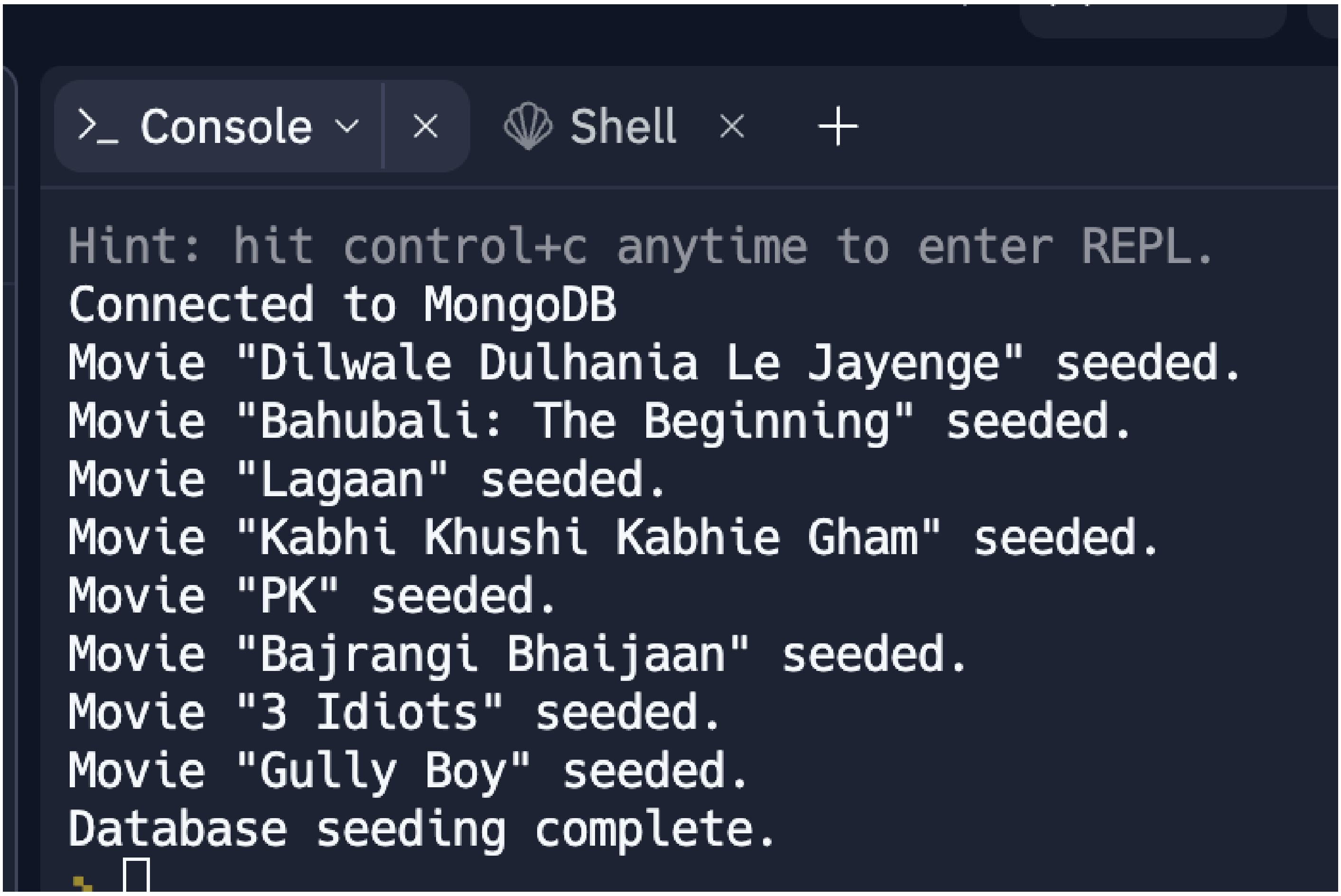
Here's a code example that demonstrates this process:

https://replit.com/@tanaypratap/BE12CW-3

```
const mongoose = require('mongoose')
const fs = require('fs')
const Movie = require('./models/movieModel') // Assuming this is the path to your movie mode

// Read JSON file
const jsonData = fs.readFileSync('movies.json', 'utf8')
const moviesData = JSON.parse(jsonData)

// Connect to MongoDB
// If you have already done this in db.js, then just require('./db') in this file.
mongoose
  .connect(mongoURI, {
    useNewUrlParser: true,
    useUnifiedTopology: true,
  })
  .then(() => {
    console.log('Connected to MongoDB')
  })
  .catch((error) => {
    console.error('Error connecting to MongoDB:', error)
  })

// Define Mongoose models (Movie, Student etc.)

// Seed Data
async function seedDatabase() {
  try {
    for (const movieData of moviesData) {
      const newMovie = new Movie({
        title: movieData.title,
        releaseYear: movieData.releaseYear,
        genre: movieData.genre,
        director: movieData.director,
        actors: movieData.actors,
        language: movieData.language,
        country: movieData.country,
        rating: movieData.rating,
        plot: movieData.plot,
        awards: movieData.awards,
        posterUrl: movieData.postedUrl,
        trailerUrl: movieData.trailerUrl,
      })

      await newMovie.save()
      console.log(`Movie "${newMovie.title}" seeded.`)
    }
    console.log('Database seeding complete.')
  } catch (error) {
    console.error('Error seeding database:', error)
  } finally {
    mongoose.disconnect()
  }
}

// Call the seedDatabase function to start seeding
seedDatabase()
```

COPY

When you run your console, your output should look like this:



```
Hint: hit control+c anytime to enter REPL.
Connected to MongoDB
Movie "Dilwale Dulhania Le Jayenge" seeded.
Movie "Bahubali: The Beginning" seeded.
Movie "Lagaan" seeded.
Movie "Kabhi Khushi Kabhie Gham" seeded.
Movie "PK" seeded.
Movie "Bajrangi Bhaijaan" seeded.
Movie "3 Idiots" seeded.
Movie "Gully Boy" seeded.
Database seeding complete.
```