

ADT Final Project Part 2

Project Title: Movie Review Portal

Team 31 members:

Akhil Reddy Vajrala

Venkata Srisai Maruthi Revanth Pasupuleti

Bharath Varma Kantheti

NodeJS Code:

```
const express=require("express");
const { default: mongoose } = require("mongoose");
const app=express()

app.listen(8000,()=>{
  console.log(`Server going on port 8000`)
});

// This is used to the mongodb.
//Authored by Bharath
address="mongodb+srv://revanth2498:revanth2498@cluster0.dh16zas.mongodb.net/?retryWrites=true&w=majority"
mongoose.connect(address).then(()=>{
  console.log("connected db using mongoose")
})
.catch(err=>{
  console.log(err.description)
})

//This is used to create user table that maintains the list of users using the website.
//Authored by Bharath
const User=mongoose.Schema({
  username:{
    type:String,
    required:true
  },
  password:{
    type:String,
    required:true
  }
});

const user=mongoose.model("Users",User)

//This is used to associate actors, directors,writers as individual persons whom can be tagged to movies tables as when one director directs or acts in multiple movies.
```

```
//Authored by Bharath
const Person=mongoose.Schema({
  name:{
    type:String,
    required:true}
});

const person=mongoose.model("Persons",Person)

//This is the review table used to store reviews.
//Author Akhil
const Review=mongoose.Schema({
  username:{
    type:String,
    required:true
  },
  movie:{
    type:String,
    required:true
  },
  year:{
    type:Number,
    required:true
  },
  review:{
    type:String
  },
  rating:{
    type:Number,
    required:true,
    min:1,
    max:10
  }
});

const review=mongoose.model("Reviews",Review)

//This is the table used to store fan theories.
//Authored by Akhil
const Fan_theories=mongoose.Schema({
  username:{
    type:String,
    required:true},
  movie:{
    type:String,
    required:true
  },
});
```

```

    year:{
      type:Number,
      required:true
    },
    fan_theory:{
      type:String
    },
    likes:{
      type:Number
    }
  });

```

```

const fan_theories=mongoose.model("Fan_theories",Fan_theories)

```

//This is the main movie table where all the reviews, fan-theories and other actors which are referenced in this table.

//Authored by Maruthi Revanth

```

const movie=mongoose.Schema({
  Title:{
    type:String,
    required:true,
  },
  Year:{
    type:Number,
  },
  Rated:{
    type: String,
  },
  Released:{
    type:String,
    required:true,
  },
  Runtime:{
    type:String,
    required:true
  },
  Genre:{
    type:String,
    required:true
  },
  Director: [{
    type: mongoose.Schema.Types.ObjectId,
    ref: 'Persons'
  }],
  Writer: [{
    type: mongoose.Schema.Types.ObjectId,
    ref: 'Persons'
  }],
});

```

```

    }],
    actors: [{
      type: mongoose.Schema.Types.ObjectId,
      ref: 'Persons'
    }],
    language:{
      type:String,
      required:true
    },
    Country:{
      type:String,
    },
    Released_Status:{
      type:String,
    },
    reviews: [{
      type: mongoose.Schema.Types.ObjectId,
      ref: 'Reviews'
    }],
    fan_theories: [{
      type: mongoose.Schema.Types.ObjectId,
      ref: 'Fan_theories'
    }],
  })

const movies=mongoose.model("movies",movie)

```

```

//Unreleased movie to check fan-made theory functionality.
//Authored by Maruthi Revanth
const unreleasedMovie = new movies({
  Title: "Loki 5",
  Year: 2024,
  Rated: "U/A",
  Released: "16-Oct-24",
  Runtime: "142 min",
  Genre: "Fantasy, Scifi",
  language: "English",
  Country: "USA"
});

unreleasedMovie.save()
  .then(
    console.log("saved movie")
  )

```

```

    ).catch(err => {
        console.error('Error saving movies:', err);
    });

    //Creating persons who will be associated with the movie Loki 5 unreleased.
    //Authored by Maruthi Revanth
    const person1 = new person({
        name: 'Tom Hiddleston',
    });

    const person2 = new person({
        name: 'Loki Director',
    });

    const person3 = new person({
        name: 'Loki Writer',
    });

    person1.save()
        .then(person => {
            console.log('Person 1 saved:', person);
            attachActor(person)
        }).catch(err => {
            console.error('Error saving person:', err);
        });

    person2.save()
        .then(person => {
            console.log('Person 2 saved:', person);
            attachDirector(person)

        }).catch(err => {
            console.error('Error saving person:', err);
        });

    person3.save()
        .then(person => {
            console.log('Person 3 saved:', person);
            attachWriter(person)

        }).catch(err => {
            console.error('Error saving person:', err);
        });

    //Authored by Maruthi Revanth
    async function attachDirector(person) {

```

```

        try {
            const movieUpdated=await movies.findOneAndUpdate({Title: "Loki 5"},
                {$push: {Director: person._id}}, {returnOriginal: false})
            console.log(movieUpdated)
        } catch (error) {
            console.error('Error associating director:', error);
        }
    }
}

//Authored by Maruthi Revanth
async function attachWriter(person) {
    try {
        const movieUpdated=await movies.findOneAndUpdate({Title: "Loki 5"},
            {$push: {Writer: person._id}}, {returnOriginal: false})
        console.log(movieUpdated)
    } catch (error) {
        console.error('Error associating writer:', error);
    }
}

//Authored by Maruthi Revanth
async function attachActor(person) {
    try {
        const movieUpdated=await movies.findOneAndUpdate({Title: "Loki 5"},
            {$push: {actors: person._id}}, {returnOriginal: false})
        console.log(movieUpdated)
    } catch (error) {
        console.error('Error associating actor:', error);
    }
}

//Authored by Akhil
const fanthoery = new fan_theories({
    username: 'user4',
    movie: 'Loki 5',
    year: 2024,
    fan_theory: 'This movie is about Loki time travel and explores how this is related
to marvel field',
    likes:7
});

//Authored by Akhil
fanthoery.save()
    .then(fan => {
        savedTheory = fan;
        console.log('Fan theory saved:', savedTheory);
        fan_theoryAttachMovie(savedTheory)
    })

```

```

    }).catch(err => {
      console.error('Error saving fantheories:', err);
    });

//Authored by Akhil
async function fan_theoryAttachMovie(fan) {
  try {
    const movieUpdated=await movies.findOneAndUpdate({Title: fan.movie},
      {$push: {fan_theories: fan._id}}, {returnOriginal: false})
    console.log(movieUpdated)
  } catch (error) {
    console.error('Error associating fantheories with movies:', error);
  }
}

//These are the reviews that we want to attach to the Inception which was already
released.
//Authored by Akhil
const review1 = new review({
  username: 'user3',
  movie: 'Inception',
  year: 2010,
  review: 'Crazy movie',
  rating:9.5
});

//Authored by Akhil
const review2 = new review({
  username: 'user2',
  movie: 'Inception',
  year: 2010,
  review: 'Great film! very very good nice nice'
});

//Authored by Akhil
async function reviewAttachMovie(review) {
  try {
    const movieUpdated=await movies.findOneAndUpdate({Title: review.movie},
      {$push: {reviews: review._id}}, {returnOriginal: false})
    console.log(movieUpdated)
  } catch (error) {

```

```

        console.error('Error associating reviews with movies:', error);
    }
}

//Authored by Akhil
review1.save()
    .then(review => {
        console.log('Review saved:', review);
        reviewAttachMovie(review)
    }).catch(err => {
        console.error('Error saving reviews:', err);
    });

//Authored by Akhil
review2.save()
    .then(review => {
        console.log('Review saved:', review);
        reviewAttachMovie(review)
    }).catch(err => {
        console.error('Error saving reviews:', err);
    });

//Query to get fan-made theories for the Loki 5.
//Authored by Maruthi
const getMovieFanTheories = async () => {
    try{
        const mo=await movies.find({Title:"Loki 5"}).populate('fan_theories')
        mo.map(movie => {
            console.log("Fan Theories:");
            movie.fan_theories.forEach(fanTheory => {
                console.log("Fan Theory:", fanTheory);
            });
        });
    }
    catch (err){
        console.log(err);
    }
}

getMovieFanTheories();

//Query to get reviews for the movie 'Inception'
//Authored by Maruthi
const getMovieReviews = async () => {
    try{
        const mo=await movies.find({Title:"Inception"}).populate('reviews')

```



```

        mo.map(movie => {
            console.log("Reviews");
            movie.reviews.forEach(review => {
                console.log("Review", review);
            });
        });
    }
    catch (err){
        console.log(err);
    }
}
getMovieReviews();

//Get movie director for the movie 'Loki 5'
//Authored by Maruthi
const getMovieDirector = async () => {
    try{
        const mo=await movies.find({Title:"Loki 5"}).populate('Director')
        mo.map(movie => {
            console.log("Directors:");
            movie.Director.forEach(director => {
                console.log("Director",director );
            });
        });
    }
    catch (err){
        console.log(err);
    }
}
getMovieDirector();

//Get movie actor for the movie 'Loki 5'
//Authored by Maruthi
const getMovieActor = async () => {
    try{
        const mo=await movies.find({Title:"Loki 5"}).populate('actors')
        mo.map(movie => {
            console.log("Actors:");
            movie.actors.forEach(actor => {
                console.log("Actor",actor );
            });
        });
    }
    catch (err){
        console.log(err);
    }
}
getMovieActor();

```

```
//Get movie writer for the movie 'Loki 5'
//Authored by Maruthi
const getMovieWriter = async () => {
  try{
    const mo=await movies.find({Title:"Loki 5"}).populate('Writer')
    mo.map(movie => {
      console.log("Writer:");
      movie.Writer.forEach(writer => {
        console.log("Writer",writer );
      });
    });
  }
  catch (err){
    console.log(err);
  }
}
getMovieWriter();
```