**Harry Potter World Project Documentation**

**Overview**

The Harry Potter World project is a web application that allows users to explore magical characters, creatures, and potions. It supports user authentication and role-based access control to manage content.

**Features**

* **User Registration and Login**: Securely register and log in to access the platform.
* **Role-Based Content Management**: Admins and editors can create, edit, and delete posts.
* **Dynamic Content Rendering**: Explore magical characters, creatures, and potions.
* **Responsive Design**: Built using EJS templates and styled with CSS.

**Project Structure**

harry-potter-world/

├── app.js # Main application file

├── config/

│ └── db.js # MongoDB configuration

├── controllers/ # Business logic for routes

│ ├── authController.js

│ └── mainController.js

├── middleware/ # Middleware for role-based access control

│ └── authMiddleware.js

├── models/ # Mongoose schemas

│ ├── itemModel.js

│ └── userModel.js

├── public/ # Static files

│ ├── css/

│ │ └── styles.css

│ └── js/ # (Empty, for custom scripts if needed)

├── routes/ # Route definitions

│ ├── auth.js

│ └── main.js

├── views/ # EJS templates

│ ├── index.ejs

│ ├── characters.ejs

│ ├── creatures.ejs

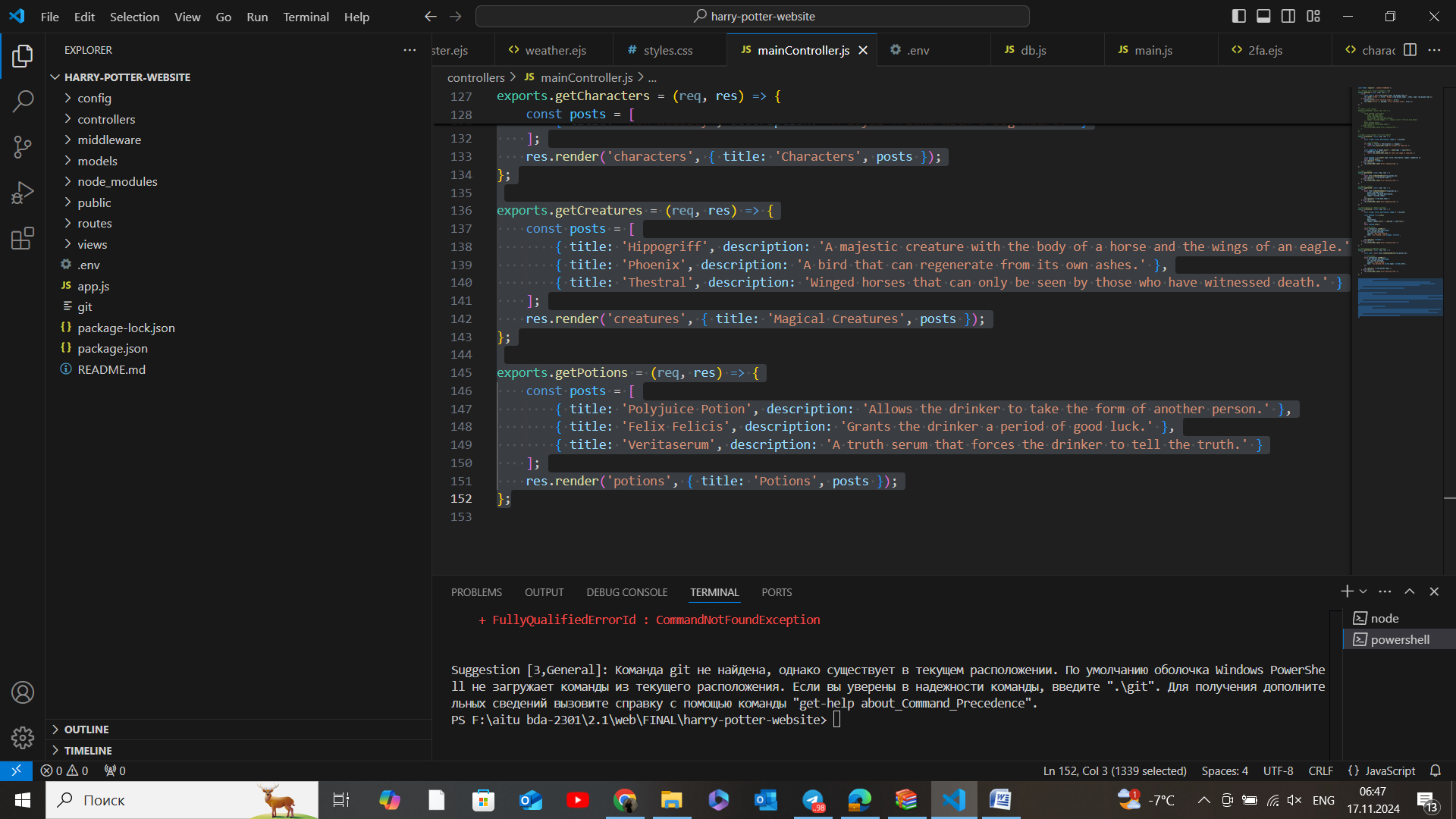
│ ├── potions.ejs

│ └── ... (other pages)

├── .env # Environment variables

├── package.json # Project metadata and dependencies

└── README.md # Project documentation

**Screenshot**

**Installation**

1. **Clone the repository:**

*git clone https://github.com/your-repository.git*

*cd harry-potter-website*

1. **Install dependencies:**

*npm install*

1. **Configure environment variables in .env:**

*MONGO\_URI=your\_mongodb\_connection\_string*

*EMAIL\_USER=your\_email*

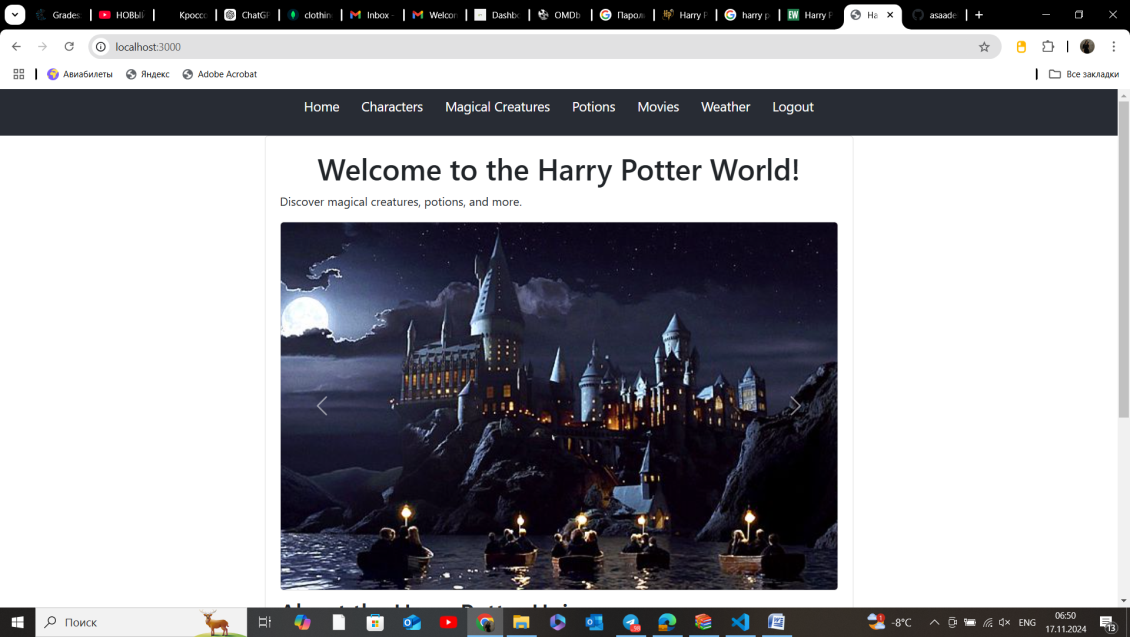
*EMAIL\_PASS=your\_password*

*SESSION\_SECRET=your\_secret*

1. **Start the server:**

*npm start*

Screenshot:

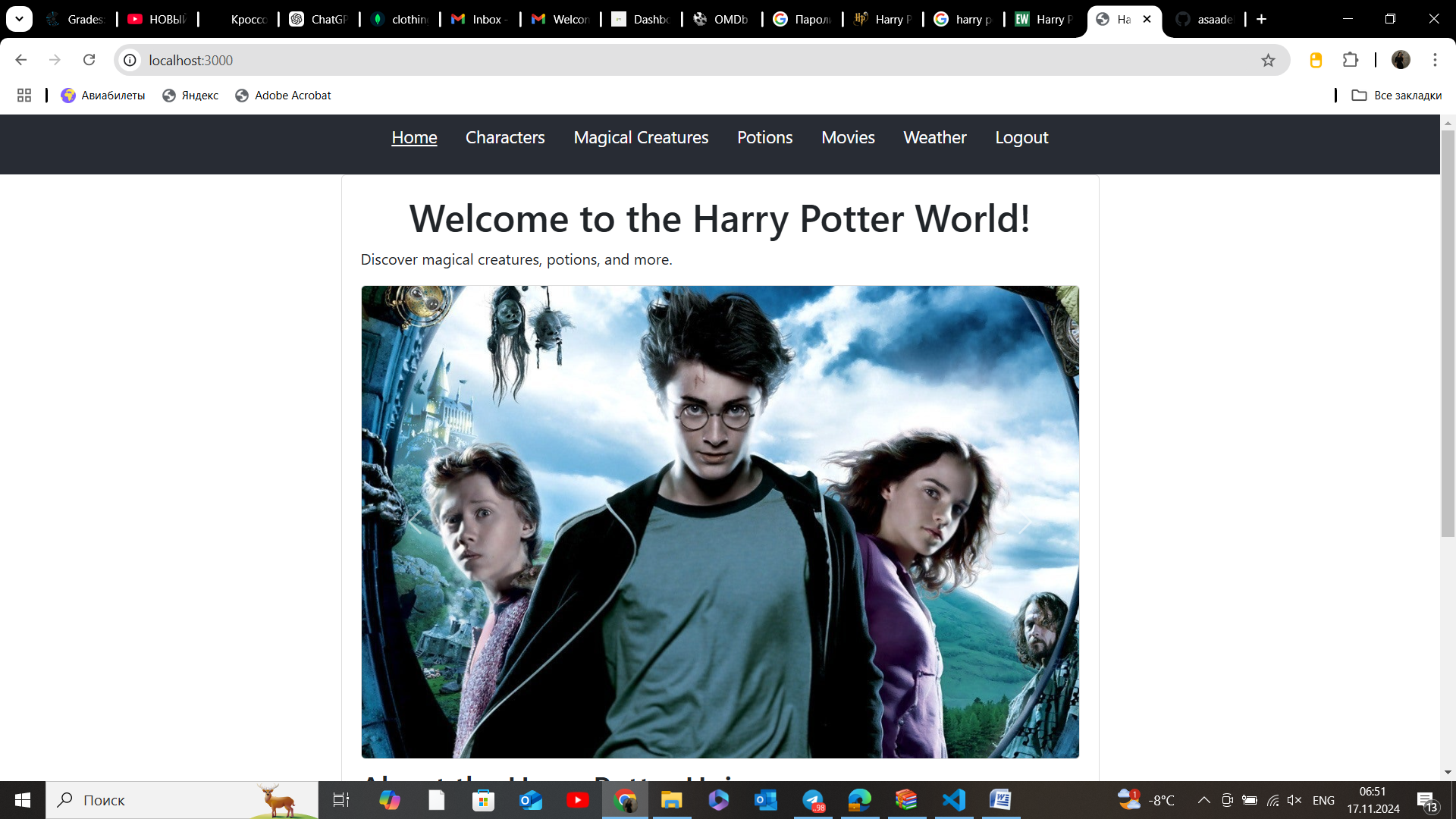


## Usage

### Homepage

The homepage introduces users to the magical world. It includes navigation links to explore characters, creatures, and potions.

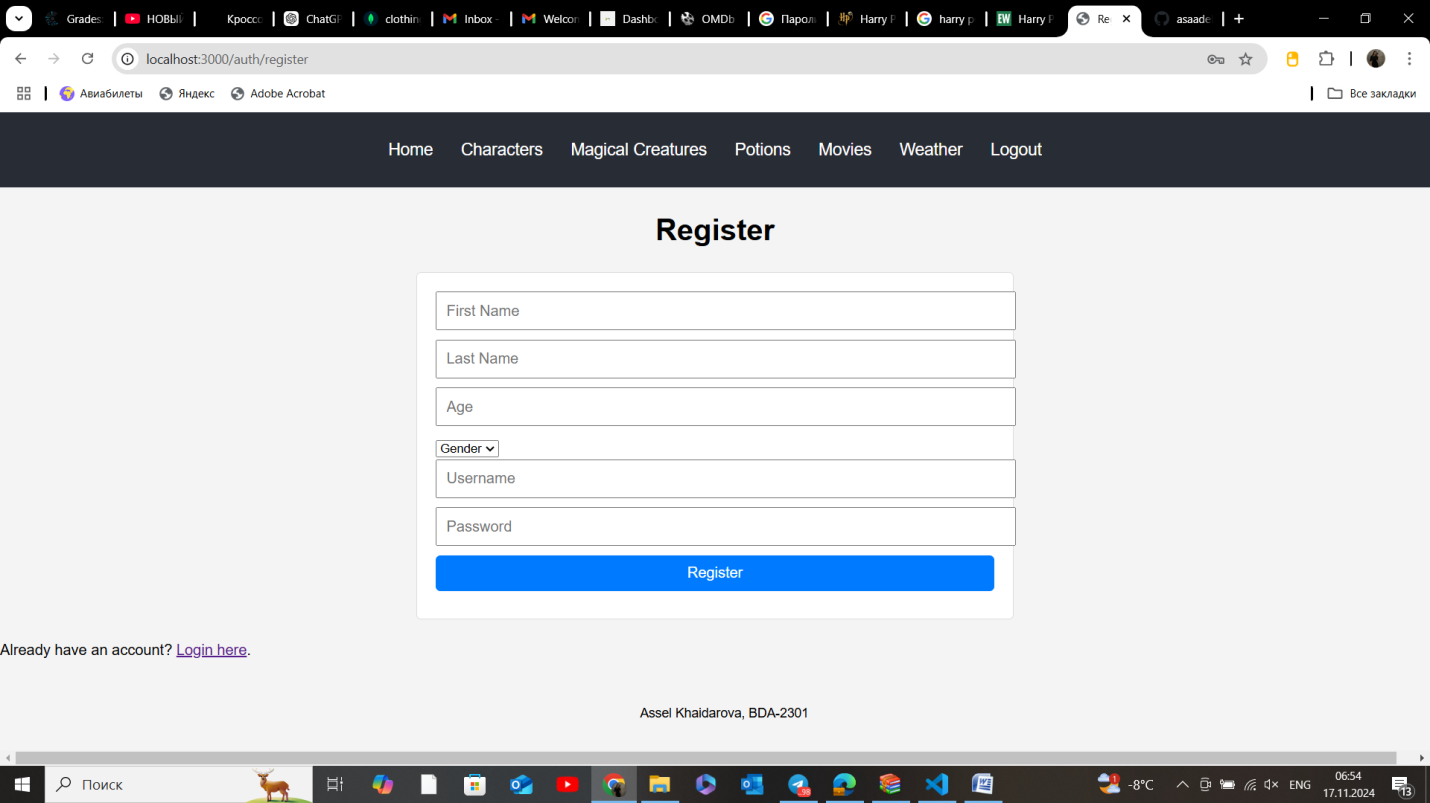
### Screenshot



### User Authentication

* **Registration Page**: Users can sign up by providing their details.
* **Login Page**: Allows access to content with appropriate permissions.

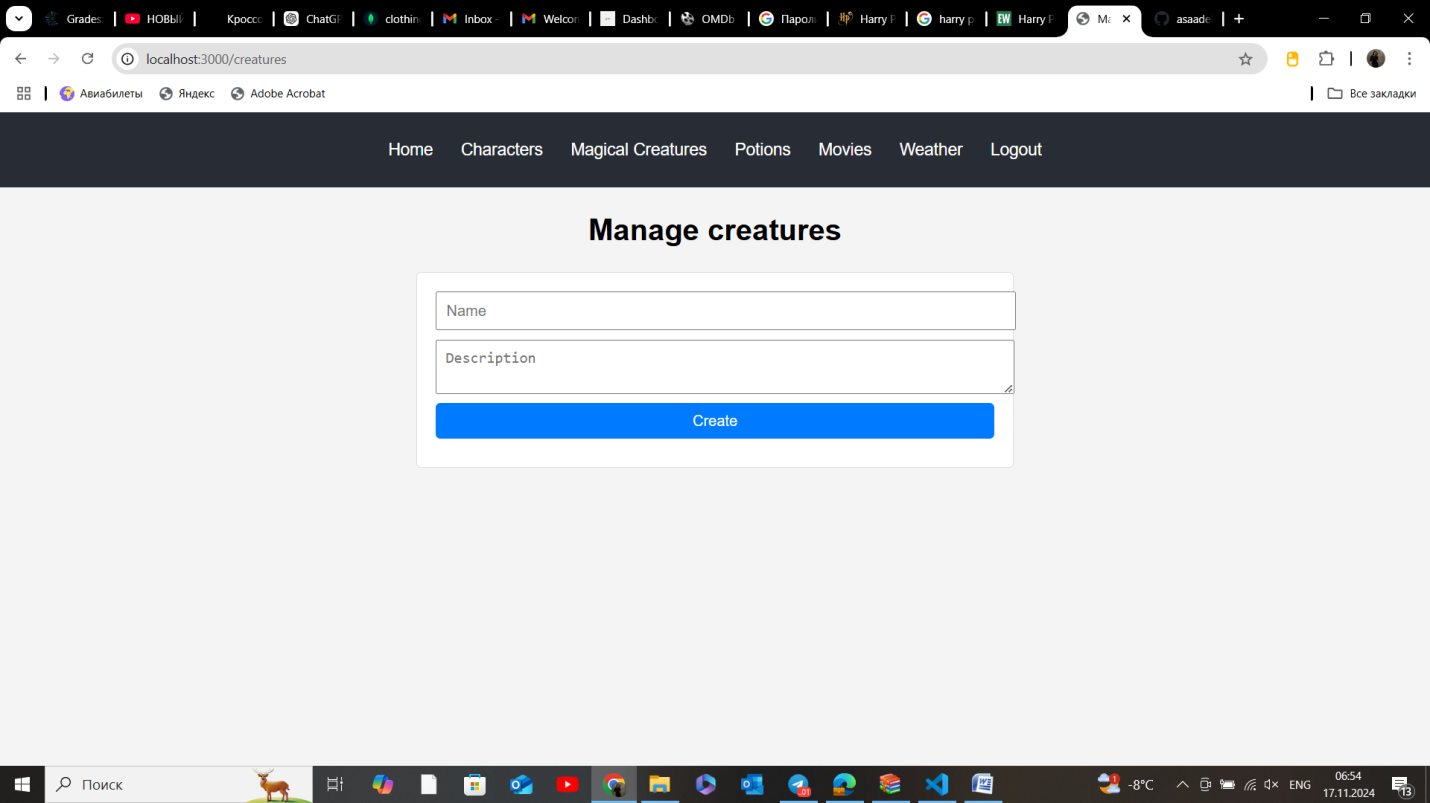
### Screenshot



### Characters, Creatures, and Potions

* **Characters**: Lists details about Harry Potter characters.
* **Creatures**: Describes magical creatures.
* **Potions**: Explains different potions and their effects.

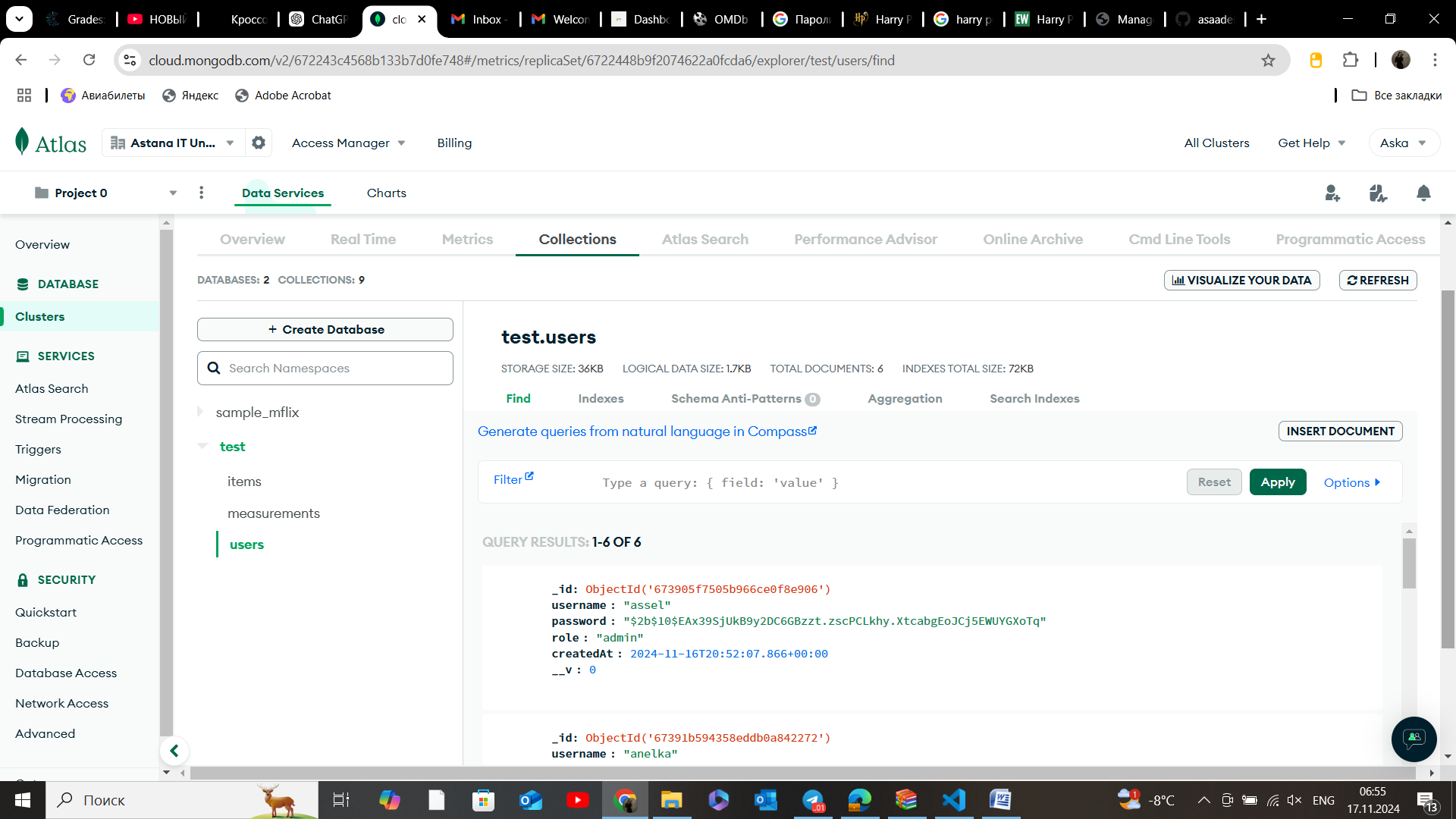
### Screenshot



## Database Integration

* **MongoDB**: Stores data for characters, creatures, potions, and user authentication.
* Data is structured using Mongoose schemas.

### Screenshot



## API and Middleware

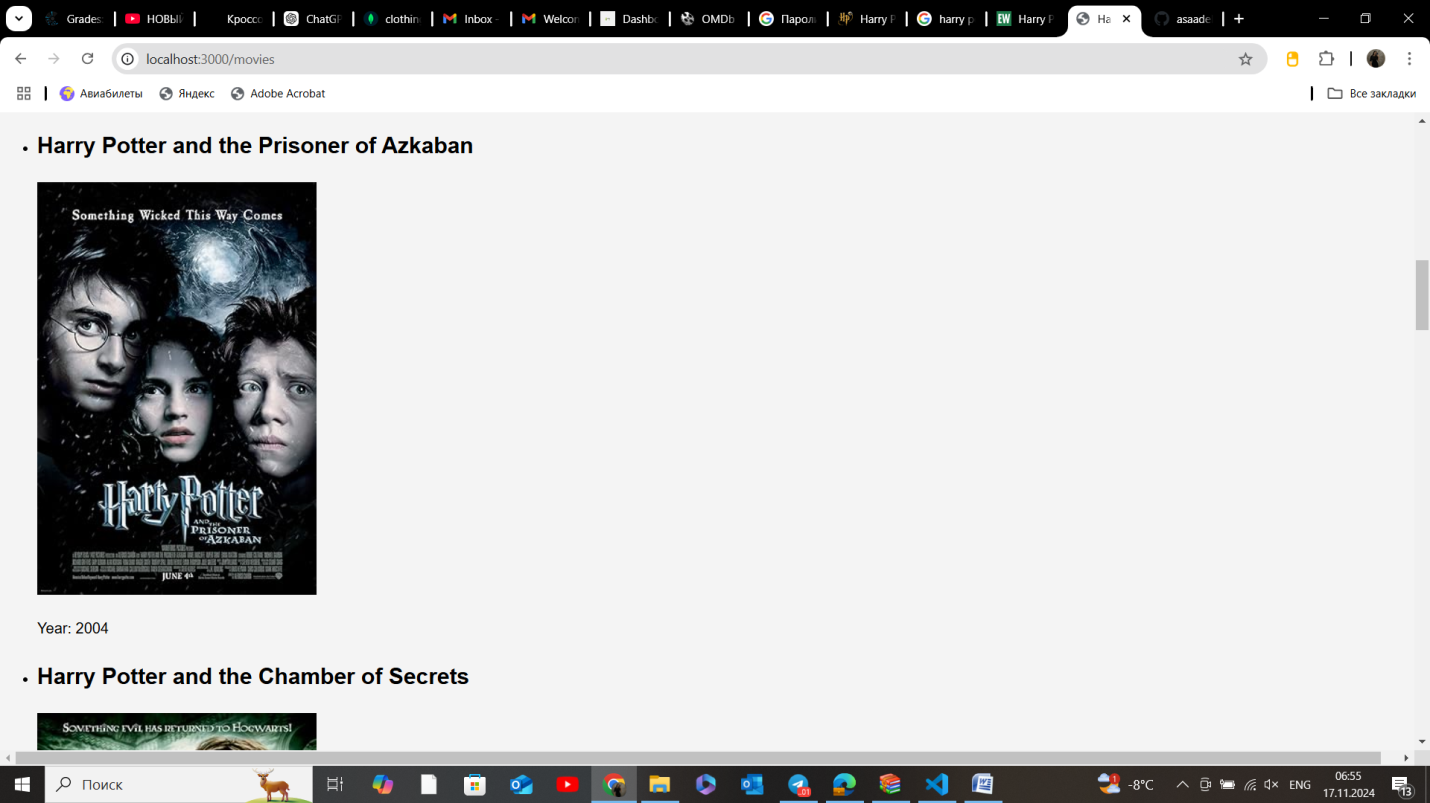
### Middleware

* authMiddleware.js ensures users have the appropriate roles for accessing routes.

### API Integration

* Weather API: Provides real-time weather updates.
* OMDB API: Fetches movie details.

### Screenshot



## Known Issues and Future Enhancements

### Known Issues

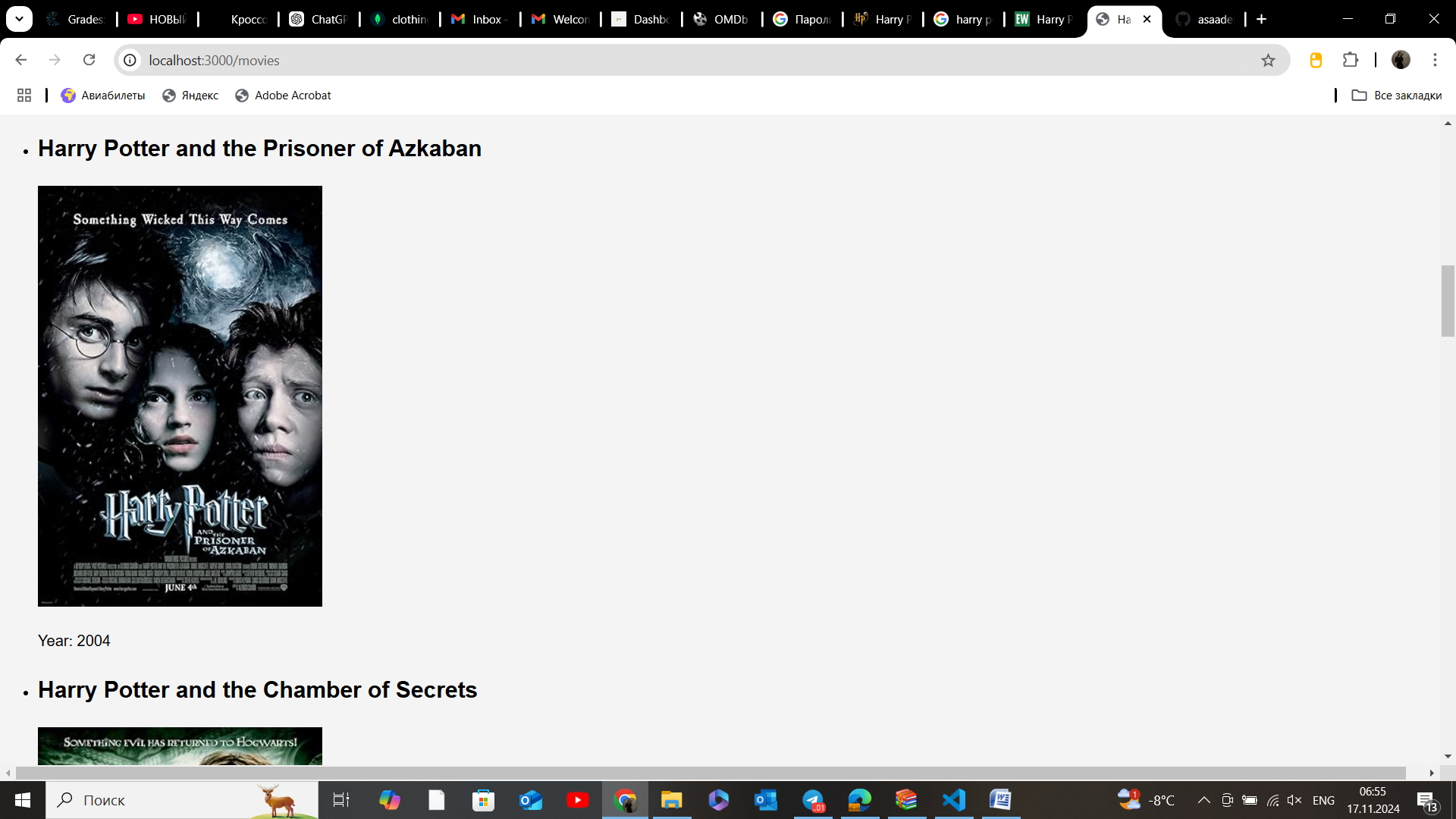
* Deprecation warnings for MongoDB options (useNewUrlParser and useUnifiedTopology).

### Future Enhancements

* Add pagination for large datasets.
* Improve UI/UX with interactive JavaScript elements.
* Expand content to include spells and locations.

## Contributors

* **Assel Khaidarova**: Developer



## License

MIT License.