12. Polling App

Overview:

The Polling App serves as an interactive platform where users can create polls on diverse topics and participate by casting their votes. Whether it's about trending topics, social issues, or even fun debates, this platform gives users the power to voice their opinions and see collective results.

Features:

1. **User Registration & Login**: The initial step for any user would be to register and log in, making sure that all interaction is authenticated.
2. **Poll Creation**: Registered users can easily create a new poll. They can set the question, multiple choice options, and even the duration for which the poll will be active.
3. **Voting**: Users can browse through the list of active polls and vote on them. The user interface should show real-time results after a user has voted.
4. **My Polls Dashboard**: Users can view a list of polls they have created with statistics like the total number of votes, leading option, etc.
5. **Results Page**: Once a poll has concluded, a results page is displayed showing the number and percentage of votes for each option.
6. **Search & Categories**: Users can search for polls based on keywords, or filter them based on categories, to enhance user experience and discoverability.

Authentication Aspects:

1. **Secure Registration**: Users must go through a secure registration process
2. **User Login**: Implement JWT for a robust and secure login.
3. **Authenticated Poll Creation**: Only registered and authenticated users should have the ability to create a poll. This not only makes the activity more secure but also allows tracking and managing the polls a user has created.
4. **Vote Limitation**: Implement logic to ensure that each user can vote only once per poll. This could involve storing a record of which users have voted on each poll in the database.
5. **My Polls Dashboard Authorization**: This area of the app should be protected so that only the logged-in user can view and manage the polls they have created.
6. **Security Measures**: Implement measures to prevent unauthorized API access to vote manipulation or data retrieval.

By incorporating these features and focusing on the authentication aspects, this Polling App would offer a comprehensive and secure platform for users to create and participate in polls. Not only does it allow for user engagement, but it also ensures that all activity is authenticated, thus maintaining the integrity of the polls.

API Documentation

Endpoint	HTTP Method	Description	Parameters/Body	Response	Require Headers
`/auth/register`	POST	Register a new user	`{ "username": "string", "password": "string", "email": "string" }`	`{ "message": "User registered successfully." }`	None
`/auth/login`	POST	Log in a user	`{ "email": "string", "password": "string" }`	`{ "token": "auth_token", "message": "Logged in." }`	None
`/polls`	POST	Create a new poll (Create)	`{ "question": "string", "options": ["option1", "option2"], "duration": "time_period_in_hours", "category": ObjectId }`	`{ "message": "Poll created.", "pollId": "id" }`	`Authori Bearer auth_to`

`/polls/:pollId`	GET	Get details of a specific poll (Read)	None	`{ "poll": { ... } }`	None
`/polls/:pollId`	PUT	Update a specific poll (Update)	`{ "question?": "string", "options?": ["option1", "option2"], "duration?": "time_period_in_hours", "category?": ObjectId }` (Fields are optional for partial updates)	`{ "message": "Poll updated." }`	`Author: Bearer auth_to`
`/polls/:pollId`	DELETE	Delete a specific poll (Delete)	None	`{ "message": "Poll deleted." }`	`Author: Bearer auth_to`
`/polls/vote/:pollId`	POST	Vote on a specific poll	`{ "option": "selected_option" }`	`{ "message": "Vote recorded." }`	`Author: Bearer auth_to`
`/polls/active`	GET	Get a list of active polls	None	`{ "polls": [list_of_polls] }`	None
`/polls/mypolls`	GET	Get a list of polls created by the logged-in user	None	`{ "polls": [list_of_user_polls] }`	`Author: Bearer auth_to`
`/polls/results/:pollId`	GET	View results of a concluded poll	None	`{ "pollId": "id", "results": [votes_per_option] }`	None
`/polls/search`	GET	Search for polls based on keywords	Query Params: `?keyword=string`	`{ "polls": [matched_polls] }`	None
`/polls/category/:categoryName`	GET	Get a list of polls based on a category	None	`{ "polls": [list_of_category_polls] }`	None

Notes:

- The `auth_token` returned from the `/auth/login` endpoint should be used to authenticate requests that require user authentication. It should be included in the headers where specified.
- `time_period_in_hours` in the `/polls/create` endpoint denotes the time duration for which the poll will be active. It should be a positive integer.
- In the `/polls/vote/:pollId` endpoint, the `:pollId` is a placeholder for the actual ID of the poll the user intends to vote on.
- The results array in `/polls/results/:pollId` gives the count of votes for each option.

This updated table reflects the necessary headers for endpoints involving the creation of polls and voting.

Schema Definition

1. User

```
{
    "_id": ObjectId,           // Unique identifier for the user.
    "username": "string",
    "email": "string",
    "password": "string",
    "registrationDate": Date,
    "pollsCreated": [          // Array of ObjectIds referring to polls created by the user.
        ObjectId,
        ...
    ]
}
```

2. Poll

```
{
    "_id": ObjectId,           // Unique identifier for the poll.
    "creator": ObjectId,       // Refers to User._id.
    "question": "string",
    "creationDate": Date,
    "expiryDate": Date,
    "category": ObjectId,      // Refers to Category._id.
    "options": [               // Embedded options for the poll.
        {
            "_id": ObjectId,
            "optionText": "string"
        },
        ...
    ]
}
```

3. Category

```
{
    "_id": ObjectId,           // Unique identifier for the category.
    "name": "string",
}
```

4. Vote

```
{
  "_id" : ObjectId,
  "userId" : ObjectId,
  "pollId" : ObjectId,
  "optionId" : ObjectId,
  "voteDate" : Date
}
```

With these additions, we have integrated the essential CRUD operations for the polls in the API documentation.