ENDPOINTS

USERS SUBROUTES:

/users/signup – POST
Allows users to sign up for an account on the website. JWT token will be returned in either the session cookie or the local storage to enable further site access. Password will be hashed in the API before sending to the MongoDB database.

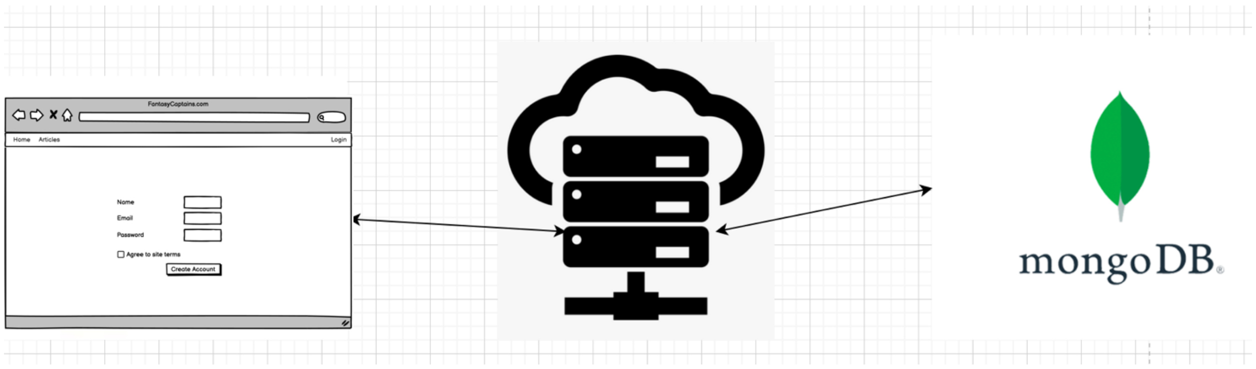
REQUEST:
Request will be a POST, and will use URL similar to:
https://fantasyprofessors.com/api/users/signup

Request body needs to include the following fields:
email: email@email.com
password: superSecret
name: John Smith

RESPONSE:
Response code 201 (successfully created):
```
{
  “userId”: abc123,
  “email”: email@email.com,
  “token”: JWT token
}
```

Response code 303 (user already exists):
```
{
  “status”: 303,
  “message”: “User already exists.”
}
```

Response code 500:
```
{
  “status”: 500,
  “message”: “There was an error creating the account.”
}
```



/users/login – POST
Allows users to login to their existing account on the website. JWT token will be returned in either the session cookie or the local storage to enable further site access. Password will be authenticated with the hashed password stored in the MongoDB database.

REQUEST
Request will be a POST, and will use URL similar to:
https://fantasyprofessors.com/api/users/login

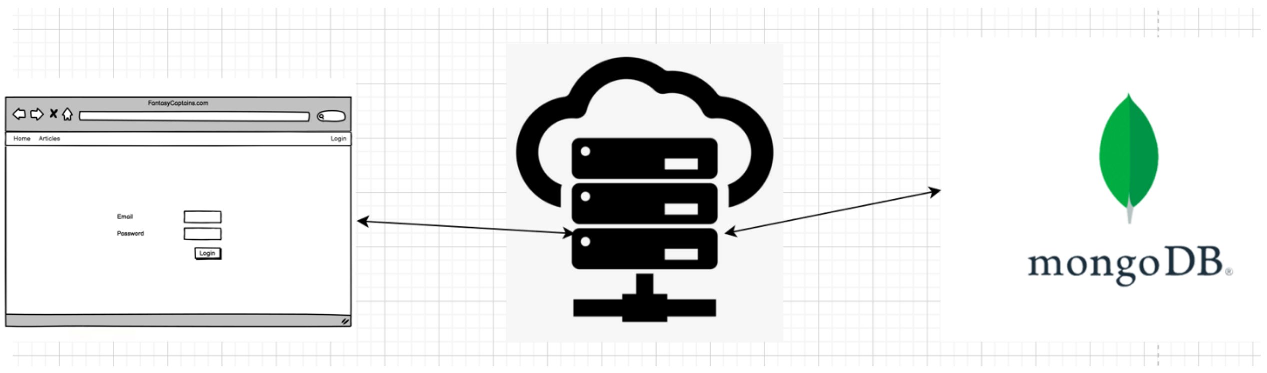
Request will need to have the following in the body:
email: email@email.com
password: superSecret

RESPONSE

Response code 200:
```
{
  “userId”: abc123,
  “email: email@email.com,
  “token”: JWT Token
}
```

Response code 403:
```
{
  “status”: 403,
  “message”: “Invalid credentials.”
}
```

Response code 500:
```
{
  “status”: 500,
  “message”: “Error logging in user.”
}
```



ARTICLES SUBROUTES:

/articles/ - GET
Allows article information to be displayed on the front-end. Will return all articles within the database.

This call will be using a URL similar to:

https://fantasyprofessors.com/api/articles. The GET method will need to be used, and the request will return a JSON object similar to the one below (there will be more than one result with the final product):

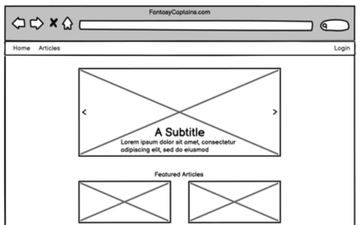
Success response:

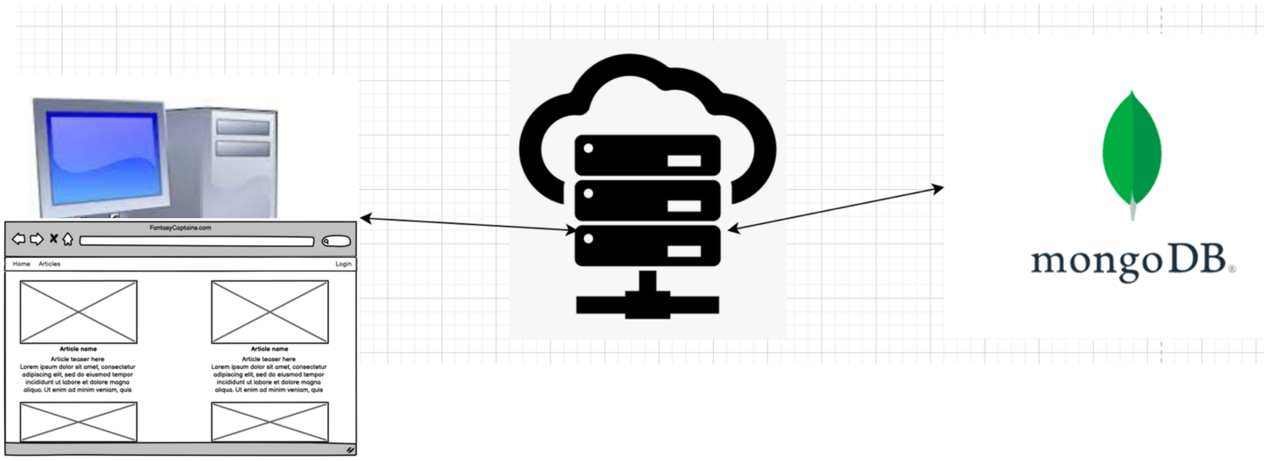
```
{
  “articles”: [
    {
      “_id”: string,
      “title”: string,
      “author”: string,
      “date”: date,
      “image”: string,
      “teaser”: string,
      “body”: string,
      “tags”: array
    }
  ]
}
```

While I do not expect this endpoint to throw errors, an example can be found below:

500 error code returned:

```
{
  “status”: 500,
  “message”: “Could not return list of articles.”
}
```


FantasyCaptains.com
Home
Articles
Login
A Subtitle
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod
Featured Articles



/articles/:aid – GET
Will be used on specific article pages where only one article is needed. Will return all information from the specific document needed, specified by the aid (article ID) variable given in the route.

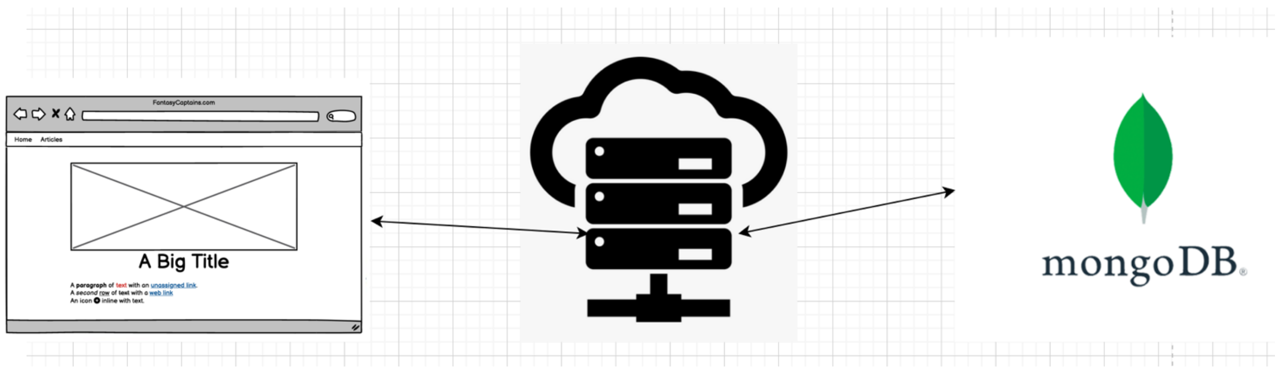
This is a GET call that will use a URL similar to: https://fantasyprofessors.com/api/articles/153
Success response:

```
{
 “articles”: [
   {
     “_id”: string,
     “title”: string,
     “author”: string,
     “date”: date,
     “image”: string,
     “teaser”: string,
     “body”: string,
     “tags”: array
   }
 ]
}
```

Example error response:

404 error code returned:

```
{
 “status”: 404,
 “message”: “Article not found.”
}
```


FantasyCaptains.com
Home
Articles
A Big Title
A paragraph of text with an unassigned link.
A second row of text with a web link
An icon inline with text.
mongoDB

/articles/create – POST

Will be used to create new articles. Authorization will be checked to ensure that user has proper credentials to make the call.

POST call, URL similar to https://fantasyprofessors.com/api/articles/create

Request should include header with JWT token:

auth: “Bearer abc123”

Body should include:

“title”: “Article Title”,
“author”: “Mason Jenkins”,
“image”: <to be determined how images will flow, may involve creating another endpoint.>,
“teaser”: “I am a teaser for an article”,
“body”: “This is a very interesting article.”,
“tags”: [“interesting”, “article”]

Success responses:

201 created

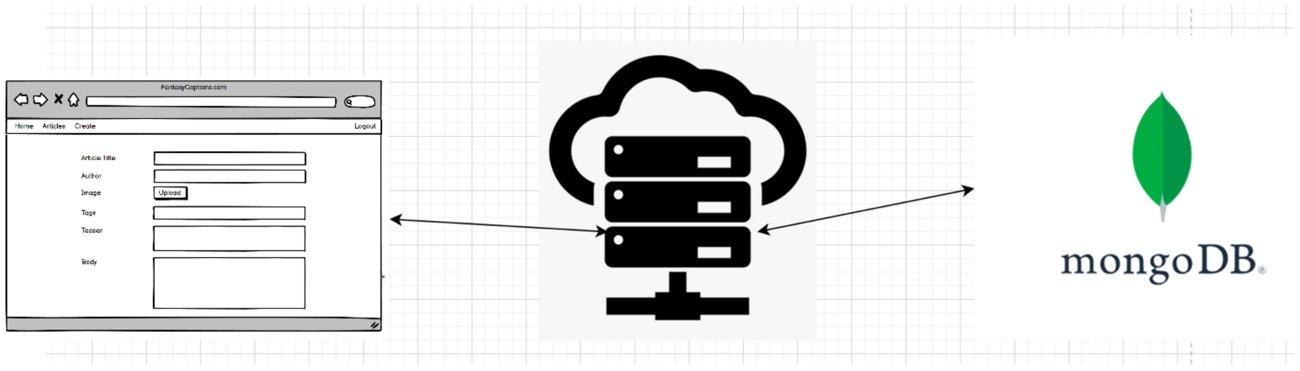
```
{
  “id”: 1234
}
```

Errors:
400 bad request

```
{
  “status”: 400,
  “message”: “Bad request.”
}
```

401 unauthorized

```
{
  "status": 401,
  "message": "Missing required authorization credential."
}
```


FantasyCaptains.com
Home
Articles
Create
Logout
Article Title
Author
Image
Upload
Tags
Teaser
Body
mongoDB