1. Users should have a first\_name, last\_name, email, password, (you can add other attributes you want to store about the user) **DONE**
2. A user should be able to sign up and sign in into the blog app. **DONE**
3. Use JWT as authentication strategy and expire the token after 1 hour **DONE**
4. A blog can be in two states; draft and published **DONE**
5. Logged in and not logged in users should be able to get a list of published blogs created **DONE**
6. Logged in and not logged in users should be able to to get a published blog **DONE**
7. Logged in users should be able to create a blog. **DONE**
8. When a blog is created, it is in draft state **DONE**
9. The owner of the blog should be able to update the state of the blog to published **DONE**
10. The owner of a blog should be able to edit the blog in draft or published state **DONE**
11. The owner of the blog should be able to delete the blog in draft or published state **DONE**
12. The owner of the blog should be able to get a list of their blogs. **DONE**
    1. The endpoint should be paginated **DONE**
    2. It should be filterable by state **DONE**
13. Blogs created should have title, description, tags, author, timestamp, state, read\_count, reading\_time and body. **DONE**
14. The list of blogs endpoint that can be accessed by both logged in and not logged in users should be paginated,
    1. default it to 20 blogs per page.  **DONE**
    2. It should also be searchable by author, title and tags. **DONE**
    3. It should also be orderable by read\_count, reading\_time and timestamp
15. When a single blog is requested, the api should return the user information(the author) with the blog. The read\_count of the blog too should be updated by 1 **DONE**
16. Come up with any algorithm for calculating the reading\_time of the blog.
17. Write tests for all endpoints
18. Create an ERD for entities and show relationships **DONE**
19. Use Winston and ensure functions and processes are logged **DONE**

**Note**:

The owner of the blog should be logged in to perform actions

**Extra points for building the views(optional)**

**Database**

1. Use MongoDB

**​Data Models**

**User**

– email is required and should be unique

– first\_name and last\_name is required

– password

​

**Blog/Article**

– title is required and unique

– description

– author

– state

– read\_count

– reading\_time

– tags

– body is required

– timestamp

**Submission**

– Push your code to GitHub

– Host it on render

– Share the render link and the GitHub link

​Submission Link: [3MTT Backend Engineering – Intermediate Module Assessment](https://forms.gle/1PCSGjSoaX7rr7MN6)

​

**Helpful links**

– [Express](<https://expressjs.com/>)

– [MongoDB](<https://www.mongodb.com/>)

​

Best of luck!