

Backend – Node.js: 100 Marks

1. Make authentication process by JWT: **(25 marks)**
 - a. SignUp
 - b. Login
2. Gallery API: **(50 marks)**
 - a. Upload Image API: User can upload images with below data
 - i. title (title for the image)
 - ii. description (about image)
 - iii. image_by (author of the image)
 - iv. is_private (identifier to know if image is public or private)
 - b. All images API
 - i. User can have the list of all public images
 - c. My Images API
 - i. User can get the list of all his/her images
 - d. Update image API
 - i. User can update his/her images
 - e. Delete image API
 - i. User can delete his/her images
3. Deployment is must: **(10 Marks)**
4. You can implement either mongoDB or psql: **(15 Marks)**

*Do not save image as text in the database, -ve marks for that.

**Marks can also be deducted while evaluation on the basis of Q&A

*Make use of these API in frontend

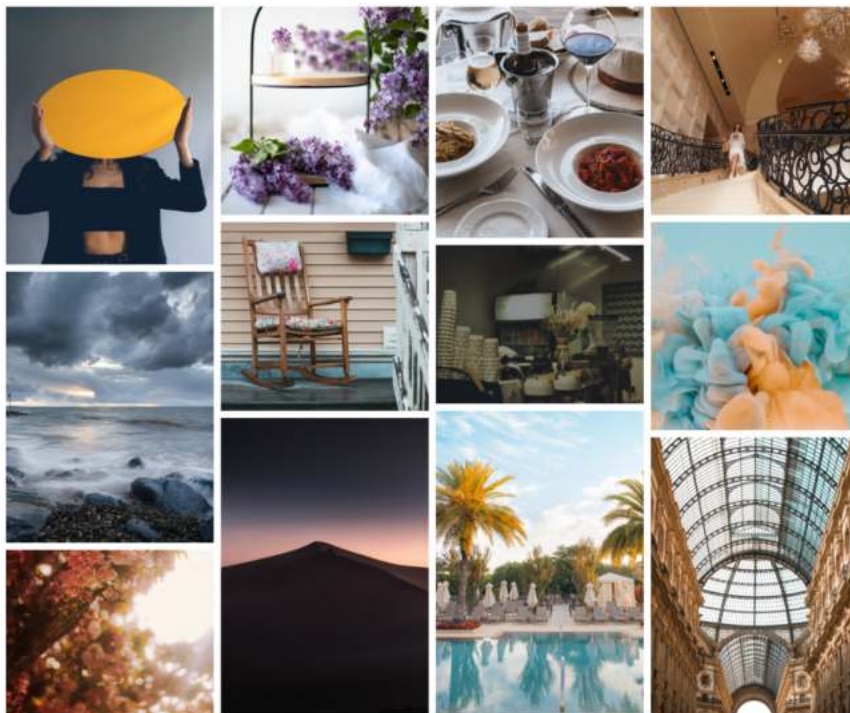
Frontend – (HTML- CSS – React): 100 Marks

1. Make a landing page which will have all the public images.
2. Make a login page. **(10 Marks)**
3. Make a signup page. **(10 Marks)**
4. When person logs in a new tab must appear on the nav bar “My Images”
5. Make use of Backend APIs
6. Deployment is must. **(10 Marks)**
7. Make proper use of Redux. **(20 Marks)**

Nav Bar will be there and it will have **(10 Marks)**

1. All, Login and SignUp when not signed in
2. All, My Images and Logout when signed in

Landing page's and My Image's body will look something like below image: **(20 Marks)**



On clicking on the image a modal should open and view is as follows: **(20 Marks)**



****Marks can be deducted if features are missing or while evaluation on the basis of Q&A**