**Admin Schema**

**1. Create a new Admin (gql)**

**Client Schema**

**1. Create a new Client (gql)**

**2. Delete a client by slug (gql)**

**3. Client can see only his activity by year. (rest)**

**4. Update client data by slug (gql)**

**5. Client can see only his published or draft blog (gql)**

**User Schema**

**1. Upload or update logged in user profile picture and delete existing one from server. (gql)**

**2. Upload or update logged in user cover picture and delte existing one from server. (gql)**

**3. User can see his profile (rest)**

**4. Login user with email and password (rest)**

**5. Update logged in user password. (rest)**

**6. Rewrite a logged in user password and verify it. (rest)**

**7. forgot Password part - 1. Take the registered phone number and sent an OTP in the phone as a text message (rest)**

**8. forgot Password part - 2. Take the OTP and match with the database one if match it will response back e boolean.**

**9. forgot Password part - 3. Last part it will take new password input and update the password.**

**Blog**  **Schema**

**1. Get blog by -- (gql)**

**Query by –**

* **Recent Blog**
* **Top of this month** 
  + **Query by viewer count of current month**
* **Most read blog**
  + **By viewer count all time**
* **All featured blog**
* **By main category**
* **If no query then show all blog**

**Extra Fetcher –**

* **Search 🡺** 
  + **By main category**
  + **By Sub Category**
  + **By Title**
  + **By keyword**
    - **Search input will have to perfectly match**
* **Pagination Include**
* **Filter by =>** 
  + **Sub Category**
  + **Publish Year**
* **Data limit will provide dynamically If not provide then will give all found data**
* **Sorted by –**
  + **Latest**
  + **A – Z** 
    - **By title name in Ascending order**
  + **Z – A** 
    - **By title name in descending order**
  + **View** 
    - **In Descending Order**

**2. Get Two Top of the month blog category by viewers and then get single top blog from each category of the current month. (gql)**

**3. Mark Blog as featured (gql)**

**4. Count a new view if a user click on individual blog page and stay their at least 60 sec (rest)**

**5. Get the all amount of blog available of all categories in the website or official schema (gql)**

**6. Get all sub category name available from what blog categories blog we want to get . (gql)**

**7. Get Individual published blog by slug with related blogs also -- (gql)**

* **Related blog will be query by** 
  + Select blog category
  + Select Blog sub category
  + Keyword

**8. Save Blog (gql)**

* **If user creates new blog then create new one and store data in temporary space**
* **If User updates and save, then only store the new data into existing one that time we should have blog id to find that blog**
* **During update time it will return then updated blog also**

**9. Published a blog (gql)**

* **If user creates and direct publish then blog will be create and publish and store content both in temporary or draft and published section**
* **If user published later in the draft section then only update the published section.**

**10. Preview a blog (gql)**

* **It generally give individual blog by id** 
  + **It will query blog content from draft section**

**11. Delete an individual blog by blog id (gql)**

**Official**  **Schema**

**1. Create an official schema (gql)**

**2. Get all Official schema’s every info (gql)**

**3. Create available main category with color (gql)**

**4. Create available sub category (gql)**

**5. Get all main category in ascending order (gql)**

**6. Get all sub category in ascending order (gql)**

**7. Add contact social media to contact officially (gql)**

**8. Update contact social media to contact officially (gql)**

**9. Delete contact social media by site name (gql)**

**Comment**  **Schema**

**1. Create a new comment or reply in the time of reply you need to pass parent comment id (gql)**

**2. Update comment by comment id (gql)**

**3. Get all comment by blog id name (gql)**

**Contact Schema**

**1. Create a new contact (gql)**

**2. Get all contact (gql)**

**3. Get individual contact by id (gql)**

**4. Reply to a contact message by provided email (gql)**