To Run the application, the docker environment has to be set up,
4 environment variables need to be present in the docker-compose.yml file

- JWTSECRET
- MONGOURI
- TZ
- USEREMAIL
- EMAILPASSWORD

As the app also relies on the GCP bucket service, a serviceKey.json has to be created and placed in the folder called config named as "serviceKey.json"

API Keys Documents

There are total of 3 routes in the backend services

1. Auth
2. Categories
3. Product

Auth

- /api/auth/register:POST
  - Parameters: name, email, contact
  - Returns: {status:true,message:Please check your email to verify your account}
- /api/auth/login:POST
  - Parameters: email, password
  - Returns: User & jwtToken
- /api/auth/me:GET
  - Headers: Authorization: jwtToken
  - Returns: User Object
- /api/auth/reset-password:POST
  - Parameters: email
  - Returns: status: true and message

Categories

- /api/categories:GET
  - Returns: categories

Products

- /api/product:POST
  - Headers: Authorization: jwtToken
  - Parameters: name, description, category, price, images
  - Returns: Product Object
- /api/product/shop:GET
  - Query Parameters:
    - search: String
    - minPrice: String
    - maxPrice: String
    - sortBy: latest || price

      - if(price)
      - orderBy: asc||desc
    - category: categoryId separated by commas
    - limit: int
    - page: int
  - Returns: Product Object Array (name, description, images, price, createdBy)
- /api/product:GET
  - Headers: Authorization: jwtToken
  - Returns Product Object Array

- /api/product/id/:id:GET
  - Headers: Authorization: jwtToken
  - Returns Single Product Object
- /api/product/id/:id:DELETE
  - Headers: Authorization: jwtToken
  - Returns Single Product Object
- /api/product/id/:id:PATCH
  - Headers: Authorization: jwtToken
  - Parameters: name, description, price, category, images
  - Returns Single Product Object