

Assignment Category: **Assignment-09_category_Orchid**

 **update: All the updates will be declared here.**

Requirements

[Requirement Video](#)

ToyTopia – A Local Kids Toy Store Platform

Project Theme

A vibrant and playful online marketplace for kids' toys, encouraging families to discover and support local toy sellers.. Users can log in, browse toys, view detailed info, and leave feedback or ratings for toys they want to buy for their kids.

Ensure the Following things to get 100% mark

- **GitHub Commits:** Include at least 10 meaningful commits with descriptive messages.
- **Readme.md:** Include a README file with the project name, purpose, live URL, key features, and any npm packages you have used.
- **Responsiveness:** Ensure the website is fully responsive on mobile, tablet, and desktop.
- **Environment Variables:** Secure Firebase configuration keys using environment variables.
- **Unique Design:** Create a vibrant, and clear UI with emphasis on user engagement.

- **Host your Application:** You can host deployment systems like Netlify, surge, and Firebase. As you develop a single-page application
 - ensure that the page doesn't throw any error on reloading from any routes.
 - Add authorized domain to Firebase if you use Netlify / surge
-

Main Requirements

● Layout Structure

1. **Navbar:** A navbar, which will contain the Website name, Home, and My Profile (challenges requirements 1). Active routes will be implemented on the navbar.
 - i. If a user is logged in, the user image will show on the Navbar, and when you hover over the user image, it will show the user name.
 - ii. If the user is not logged in it will show a login button.
 - iii. So if a user is logged in you will show the user image and a logout button. If you click on the logout button make sure you have logged out.
2. **Main Section:** Main Section will show different pages based on routes.
3. **Footer:** A Footer with all relevant information and eye-catching design (Links to terms and conditions, privacy policy, and social media accounts.)

💡 **Make sure the Navbar and Footer are showing on all pages without an Error/404 page.**

● JSON Data Generation

4. Create your JSON data and host the image on [imgbb/postimages](#)

Example,

```
{
  "toyId": 1,
  "toyName": "Lego Classic Bricks",
  "sellerName": "Toys R Us Local",
  "sellerEmail": "contact@toysruslocal.com",
  "price": 49.99,
  "rating": 4.7,
  "availableQuantity": 75,
  "description": "A timeless set of colorful Lego bricks that encourages creativity and problem-solving. Kids can build anything they imagine, from houses to vehicles.",
  "pictureURL":
    "https://cdn.pixabay.com/photo/2016/03/31/20/11/lego-1297582_1280.png",
  "subCategory": "Building Blocks"
},
```

- **Home**

5. **The home page will have these sections — Slider, Popular Toys, 2 extra sections**
6. **Slider:** Add a slider (Daisy UI slider or Swiper slider) with a minimum of 3 slides showing off some toys or so.
7. **Popular Toys:** In the upcoming events section you need to create a JSON minimum of 6 data. You need to show some data on the card layout on the home page.

Each event card will contain - JSON Data:

- i. thumbnail,

- ii. toyName
- iii. rating
- iv. availableQuantity
- v. Price
- vi. and a “View More” button.

(and other info what you want to show on the card is totally up to you)

8. Toy Details Page: (Protected)

When the user clicks the “View More” button it takes the user to the Toy Details page. This page will be a protected route and ensure that the private route moves to the login page if the user is not logged in. You need to show all the information about the clicked toy.

On this page, Include a **form to Try Now** toy, containing the following input fields:

- Name
- Email

And add a **“Try Now”** button to submit the form. You don’t need to store the data in localStorage or database. Just show a successful message on form submission.

💡 The user shouldn’t be redirected to the login page after you reload a page of the private route.

• Authentication

9. **Login Page:** When you click the login button on the navbar /protected routes it redirects to the login page. You have to use a password and email-based authentication to log in. The login page will have-
 - a. Email

- b. Password
- c. Google login,
- d. A link that will redirect to the Register page

10. Register Page: You have to use a password and email-based authentication to register. The Register page will have the following


-

- a. Name
- b. Email
- c. photoURL
- d. password
- e. A Link that will redirect to the login page
- f. Google Login

★ For password verification you need to follow this -

- Must have an Uppercase letter in the password
- Must have a Lowercase letter in the password
- Length must be at least 6 character

 **Show all the error and success messages by using toast/sweet alert.**

 Don't implement email verification method as it will inconvenience the examiner. If you want, you can add these after receiving the assignment result.

Other Requirements

1. **404 page:** Create a 404 page/ not found page.
2. **Extra route:** Add 1 more extra route to your website. This route should be private and protected, and its contents should be meaningful and relevant.
3. **Dynamic Title:** You need to implement a dynamic title for each page.
4. Once a user is logged in, user information will show on the navbar even if he/she reloads the website, this info will not disappear. You can show a

loader when the logged-in user info is in a loading state. You can implement this using Firebase's `onAuthStateChanged` method.

Challenges

1. **My Profile:** Create a protected route where, after logging in, users can see their information like- name, email, and photoURL and there will be a form with 2 input fields for name and photoURL. Users can edit the name and photoURL and save the changes. This "saving" process uses Firebase's [updateProfile\(\)](#) method to update the user's information stored securely in Firebase.
2. Implement show and hide passwords on the registration page. By default, it won't show the password, but when you click on the "Eye" icon, it will. In this way, you need to toggle.
3. Implement any of 1 packages from the following list:
 - a. [Swiper slider](#)
 - b. [Aos animation](#)

4. Forget Password

Make your forgot password feature functional. Clicking "forget password" redirects the user to the forget password page, containing a form with an email field and a reset password button.

- Suppose a user enters an email address on the login page. Show the email also in the forget password form.
- On clicking the reset button. Redirect the user to Gmail.

What to Submit

Your Github Repo Link :

Your Live Link :