

# REACT PROJECTS

For practicing React and improving proficiency.

By Bedirhan KURT 

# PROJECT 1

# CALCULATOR

## Strengthening React Basics

- **Project:** Build a simple “Calculator” application. Users can perform basic arithmetic operations: addition, subtraction, multiplication, and division.

- **Details:**

- Include an input field and buttons for each operation.
- Use `useState` to manage calculation results.
- Perform calculations between two numbers and display the result dynamically.

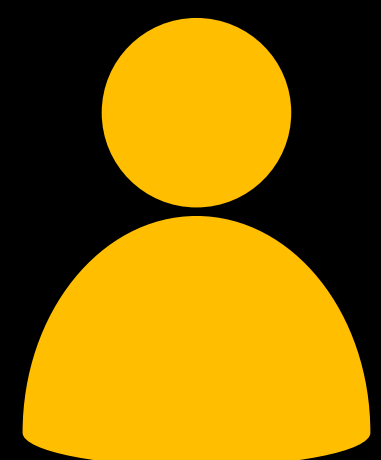


# PROJECT 2

# USER CARDS

## Props and State Management

- **Project:** “User Cards” application. Display user profiles with name, age, and profession on cards, allowing users to edit their information.
- **Details:**
  - Each user is represented by a reusable component.
  - Editing user data opens an input modal or inline fields.
  - Use props for passing data and state for dynamic updates.



# PROJECT 3

## WEATHER APP

### Working with APIs

- **Project:** “Weather App” that fetches weather data from an API (e.g., OpenWeatherMap) and displays it dynamically.
- **Details:**
  - Users input a city name to get its weather information.
  - Use fetch or axios for API calls.
  - Implement useEffect for side effects.
  - Show a loading spinner while fetching data.

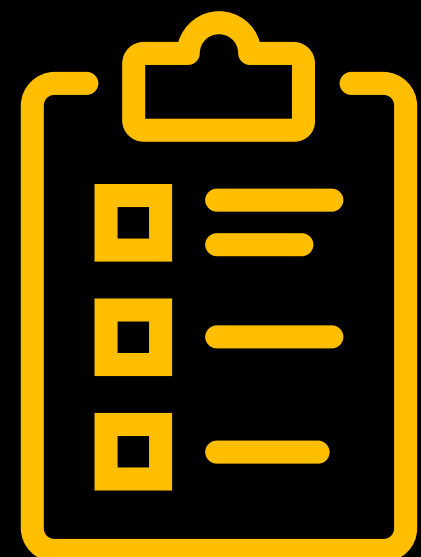


# PROJECT 4

## REGISTRATION FORM

### Forms and Validation

- **Project:** “Registration Form” where users enter their name, email, and password. The form validates inputs and displays error messages if necessary.
- **Details:**
  - Include form fields and a “Submit” button.
  - Display errors for empty fields or invalid email/password formats.
  - Create a custom hook (useForm) to manage form validation.



# PROJECT 5

## BLOG APP

### Routing with React Router

- **Project:** “Blog App” with multiple pages, including a homepage and a blog detail page.
- **Details:**
  - Use React Router to create dynamic routes.
  - Display blog titles on the homepage.
  - Clicking a title navigates to the detailed blog post using URL parameters.



# PROJECT 6

## NOTE APP

### Performance Optimization

- **Project:** “Note Taking App” that supports a large list of notes and includes search and filter features.
- **Details:**
  - Optimize rendering with `useMemo` and `useCallback`.
  - Prevent unnecessary re-renders of components.
  - Implement fast search and filtering for a better user experience.

# PROJECT 7

## IMAGE GALLERY

### Animations

- **Project:** “Image Gallery” where users can browse images with smooth animations and transitions.
- **Details:**
  - Use libraries like react-spring or framer-motion for animations.
  - Add hover effects and transition effects when switching images.
  - Enable swipe or navigation controls between images.





# PROJECT 8

## SHOPPING SITE

### Redux or Redux Toolkit

- **Project:** “Shopping Cart” for an e-commerce platform where users can add/remove items from their cart and view totals.
- **Details:**
  - Use Redux Toolkit to manage cart state globally.
  - Display item counts and total price dynamically.
  - Allow users to modify item quantities in the cart.



# PROJECT 9

## CLOCK APP

### Advanced Hooks

- **Project:** “Clock App” that displays real-time clock updates and lets users set alarms.
- **Details:**
  - Use `useReducer` to handle alarm functionality.
  - Manage references for active alarms using `useRef`.
  - Update the displayed time every second.



# PROJECT 10

## WRITING TESTS

### Writing Tests

- **Project:** Pick any previous project and write tests for it.
- **Details:**
  - Write test cases for functionality like adding tasks or displaying user data.
  - Use Jest and React Testing Library for testing.
  - Include component and user behavior tests.



# FINAL PROJECT

## RECEIPE APP

### General Practice

- **Project:** “Recipe App” where users can search for recipes, save them to favorites, and view detailed instructions.
- **Details:**
  - Fetch recipe data from an API and display it in a list.
  - Allow users to add/remove recipes from favorites.
  - Combine all previous skills, including API integration, routing, and global state management.

