**Задание #1**

**Цель:** изучение работы с state

**Описание:** Создание рабочего счетчика, состоящего из 3-ех кнопок (increment, decrement, reset) и отображаемой цифры результата.

**Требования:**

1. Создать проект с помощью **create-react-app**
2. Выложить проект на **Github Pages** (читайте документацию create-react-app)
3. Использовать **state** для хранения значения счетчика
4. Использовать следующую структуру для хранения файлов:  
   а) **containers/CounterContainer.js** – файл, представленный в виде ES6 class, содержащий всю логику счетчика (хранения и изменения), но не содержащий не одного html элемента. В return содержит лишь одну строку  
   return <Counter {…props} />  
   б) **views/Counter/index.js** – файл, представленный в виде arrow function, содержащий всю html разметку и подключает переданные props для кнопок.

в) **views/Counter/styles.js** – файл, содержащий css код, представленный в виде js (смотрите пункт 6)

1. Описать все props через библиотеку **prop-types**
2. Для верстки использовать библиотеку **material-ui** версии 1 и выше
3. Создать git tag версии v0.1.0

**Результат:** сбросить ссылки на github pages с готовым счетчиком и ссылку на github с выбранным тегом v0.1.0

**Задание #2**

**Цель:** изучение lifecycle

**Описание:** продолжаем предыдущее задание.Над счетчиком добавим счетчик счетчиков (назовем его родителем), состоящего из 3-ех кнопок (добавление нового счетчика, удаление первого счетчика и вернуться к начальному состоянию). Всегда должен оставаться минимум один счетчик. При нажатии на добавление нового счетчика, появляется еще 1 счетчик из первого задания, с начальный значением 0, а все счетчики с четным значением должны увеличиться на 1. При нажатии на удаление первого счетчика, должен пропасть последний счетчик, а все счетчики с нечетным значением, должны уменьшиться на 1.

**Требования:**

1. Продолжать следовать логике требований 2-6 задания #1
2. Для счетчиков с логикой четных/нечетных счетчиков использовать функцию **UNSAFE\_componentWillRecieveProps**
3. Отключить render компонентов у который после добавления /удаления счетчика осталось тоже значение. Использовать для этого функцию **shouldComponentUpdate**
4. Читабельно логировать для счетчиков и родителя в консоль вызов следующих методов: **componentDidMount, shouldComponentUpdate, UNSAFE\_componentWillRecieveProps, componentDidUpdate, render, componentWillUnmount, getDerivedStateFromProps, getSnapshotBeforeUpdate**
5. Изучить работу методов по логам
6. Создать git tag v0.2.0

**Результат:** сбросить ссылки на github pages с выполненным заданием и ссылку на github с выбранным тегом v0.2.0

**Задание #3**

**Цель:** знакомство с библиотекой react-router-dom

**Описание:** подключить роутер, где по урлу:

1. / – мы будем видеть 2 неактивных tabs из material-ui. Первый называется «о нас», а второй «счетчики»
2. /about – мы будем видеть активный tab «о нас» и неактивный tab «счетчики». При этом внизу будет отображаться некий отформатированный контент о нашей компании на ваш выбор
3. /counters – мы будем видеть активный tab «счетчики» и неактивный tab «о нас». При этом будет видеть страницу счетчиков из задания #2
4. Любой другой роут должен редиректиться на /404 – где не будет табов, лишь по центру отформатированный текст «404 – страница не найдена»

**Требования:**

1. Продолжать следовать логике требований 2, 4, 5, 6 задания #1
2. Создать git tag v0.3.0

**Результат:** сбросить ссылки на github pages с выполненным заданием и ссылку на github с выбранным тегом v0.3.0

**Задание #4**

**Цель:** работа с формами

**Описание:** добавить таб c название «войти» с путем /login. В этом tab реализовать форму авторизации пользователя, состоящая из полей почта, пароль и кнопки войти. Под формой должно быть выведены введенные на текущий момент значения полей обновляющиеся в реальном времени

**Требования:**

1. Продолжать следовать логике требований 2, 4, 5, 6 задания #1
2. Нельзя использовать библиотеки работы с формой
3. Должна отображаться валидация на почту и пароль минимум 6 символов
4. При успешном прохождении валидации, по нажатию enter или кнопки войти, должна в консоли появится запись с введенными данными и данные в форме очистятся
5. Для данных введенных в форму нужно применить JSON.stringify и вывести на экран
6. Создать git tag v0.4.0

**Результат:** сбросить ссылки на github pages с выполненным заданием и ссылку на github с выбранным тегом v0.4.0

**Задание #5**

**Цель:** знакомство с redux

**Описание:** добавить таб c название «войти с помощью redux» с путем /login-redux. В этом tab реализовать форму авторизации пользователя, состоящая из полей почта, пароль и кнопки войти. Под формой должно быть выведены введенные на текущий момент значения полей обновляющиеся в реальном времени

**Требования:**

1. Продолжать следовать логике требований 2, 4, 5, 6 задания #1
2. Нельзя использовать библиотеки работы с формой
3. Использовать redux store для хранения данных введенных в форму
4. Должна отображаться валидация на почту и пароль минимум 6 символов
5. Должны использовать общие функции валидации полей для задания #4, #5
6. При успешном прохождении валидации, по нажатию enter или кнопки войти, должно перейти на новую страницу /login-redux/success и вывести введенные значения в форме
7. Создать git tag v0.5.0

**Результат:** сбросить ссылки на github pages с выполненным заданием и ссылку на github с выбранным тегом v0.5.0

**Задание #6**

**Цель:** знакомство с библиотекой redux-form

**Описание:** добавить таб c название «войти с помощью redux-form» с путем /login-redux-form. В этом tab реализовать форму авторизации пользователя, состоящая из полей почта, пароль и кнопки войти. Под формой должно быть выведены введенные на текущий момент значения полей обновляющиеся в реальном времени

**Требования:**

1. Продолжать следовать логике требований 2, 4, 5, 6 задания #1
2. Хранить данные в своем redux store, только после отправки формы
3. Должна отображаться валидация на почту и пароль минимум 6 символов
4. Должны использовать общие функции валидации полей для задания #4, #5, #6
5. При успешном прохождении валидации, по нажатию enter или кнопки войти, должно перейти на новую страницу /login-redux-form/success и вывести введенные значения в форме
6. Создать git tag v0.6.0

**Результат:** сбросить ссылки на github pages с выполненным заданием и ссылку на github с выбранным тегом v0.6.0