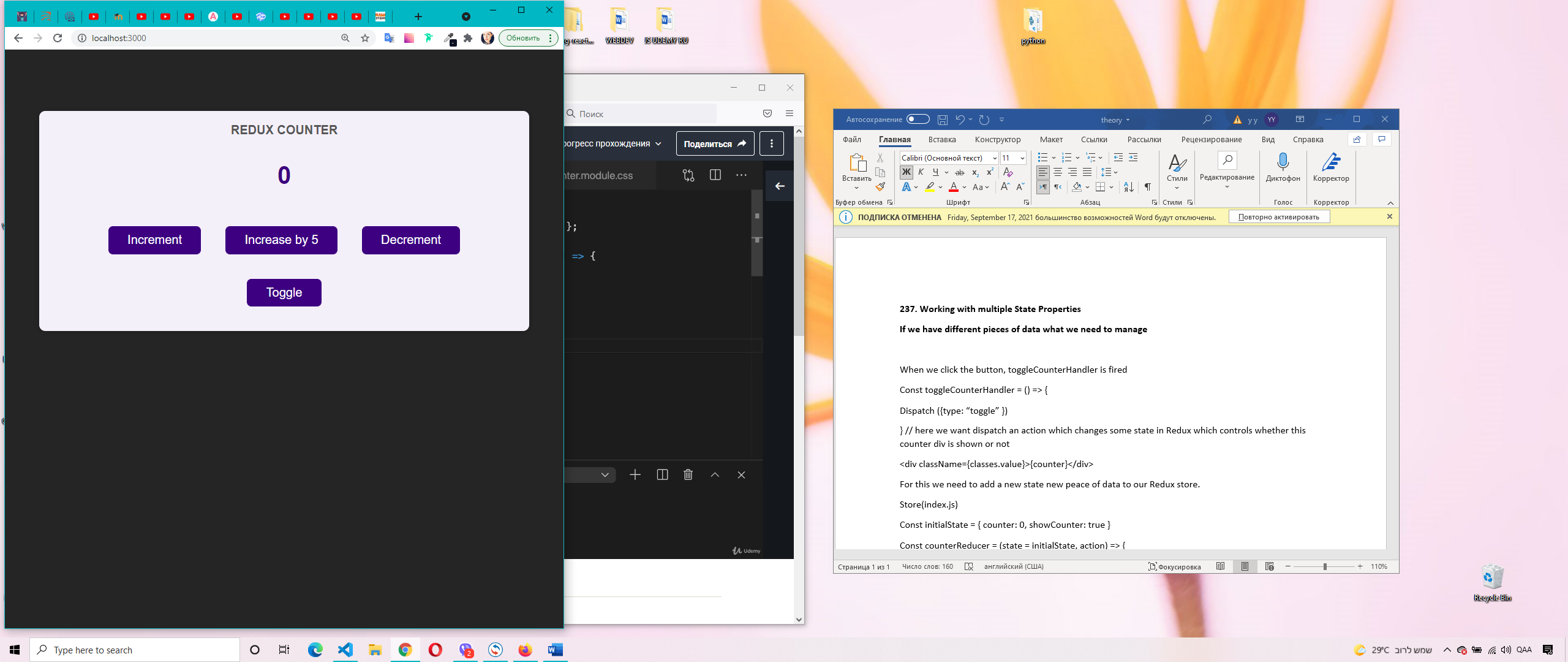
**237. Working with multiple State Properties**

**If we have different pieces of data what we need to manage**



When we click the button, toggleCounterHandler is fired

Const toggleCounterHandler = () => {

Dispatch ({type: “toggle” })

} // here we want dispatch an action which changes some state in Redux which controls whether this counter div is shown or not

<div className={classes.value}>{counter}</div>

For this we need to add a new state new peace of data to our Redux store.

Store(index.js)

Const initialState = { counter: 0, showCounter: true }

Const counterReducer = (state = initialState, action) => {

//for action type increment we don’t want to change nothing but we must save property

Counter: state.counter + 1,

showCounter: state.showCounter

if (action.type === ‘toggle’) {

return (

showCounter: !state.showCounter

counter: state.counter

hide and show the number

const show = useSelector((state) => state.showCounter);

 {show && <div className={classes.value}> {counter} </div>}

When press toggle button showCount : false and we don’t see the number.

Изображение выглядит как текст, монитор, снимок экрана, внутренний

Автоматически созданное описание

**238. How to work with Redux State Correctly**

The object what return in reducers they will overwrite the existing state. Can be side effect. For exapmle if we make

 if (action.type === "increase") {

    return {

      counter: state.counter + action.amount,

    };

  }

Instead

 if (action.type === "increase") {

    return {

      showCounter: state.showCounter,

      counter: state.counter + action.amount,

    };

  }

So showCounter = undefined = false and will be error.

YOU SHOULD NEVER MUTATE THE STATE, the existing state(the original state)

**239. Redux toolKit(can be article for the blog)**

Potential problems:

* Many types of actions in huge project
* We can have a lot of properties of state and all the time need to copy it

For example, for ensuring that we have unique identifiers and we don’t miss type we can create constans.

But we have more easy solution: library Redux ToolKit simply as an extra package which makes working with Redux more convenient and easier.

npm install @reduxjs/toolkit

240. Adding State Slices

Import {createSlice} from ‘@reduxjs/toolkit’;

Emphasize – обратить внимание

Call it the low initialState object

createSlice({}) – wants an object as an argument. We prepare a slice of our global state. And when we have peaces of the state that are not direcly related we could create different slices potentinally in different files to make our code maintainable(легкий в обслуживании).

Every slice need a name but the name is up to you, next you need to set up an initial state, and also need to add reducers(map of reducers that this slice of state needs)- 4 methods. Every method will then automatically receive the latest state.

We don’t have to write our own if checks anymore which also reducers some boilerplate code(шаблонный код)we would have to write otherwise.

Inside this method we are allowed to mutate the state. For example we can set state.counter++ for example. SEEMS to be allowed. We atill must not manipulate the existing state. But good that by using redux toolkit and functions like create slice, we can’t accidentally manipulate the existing state. Because Redux toolkit internally uses another package, called imgur, which will detect code like this and which will automatically clone the existing state, create a new state object, keep all the state which we are not editing and override the state which we are editing in an immutable way. So we still have immutable code here even though it doesn’t look like it because of this internally used package.

And therefore we as a developer have a much easier time working with Redux because we don’t have to create a copy manually and keep all the code we’re not changing, instead, we just change the code we wanna change and internally it’s translated into immutable code.

Using Redux toolkit we still have reducers that listen to actions that have ab extra payload, extra data. If we need some data that’s attached to the action then we can still accept is as a parameter and use it in the reducer function the the reducer method.

Writing this code is certainly quite convenient and shorter but how do we now make our store aware of that slice?(в курсе.осведомленный). How do we use that slice?