Plan for implementation Firebase into Restaurant project:

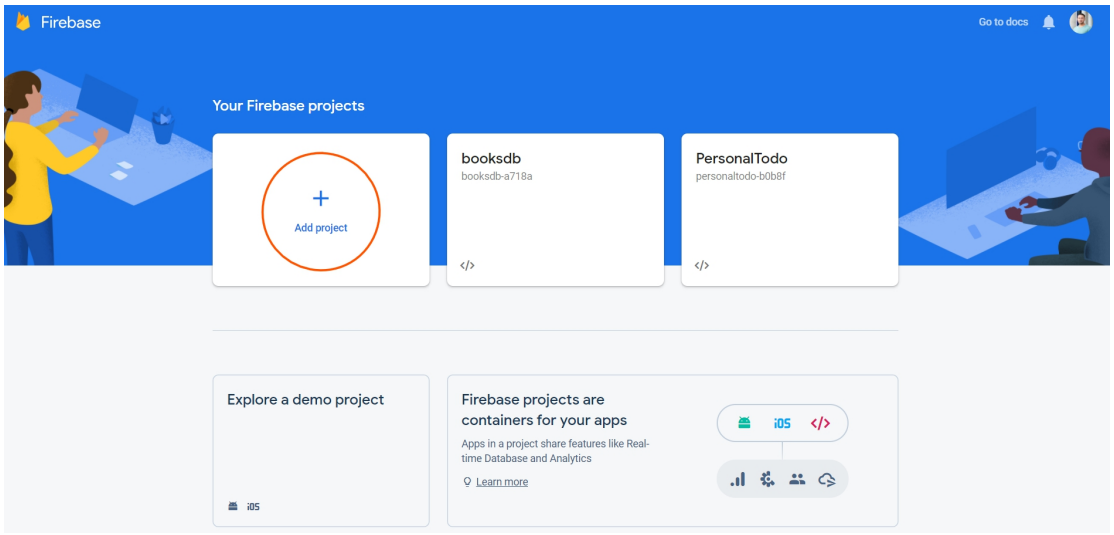
Download zip file via Code button: https://github.com/DonskiLive/71-10_11-Home-Task-71

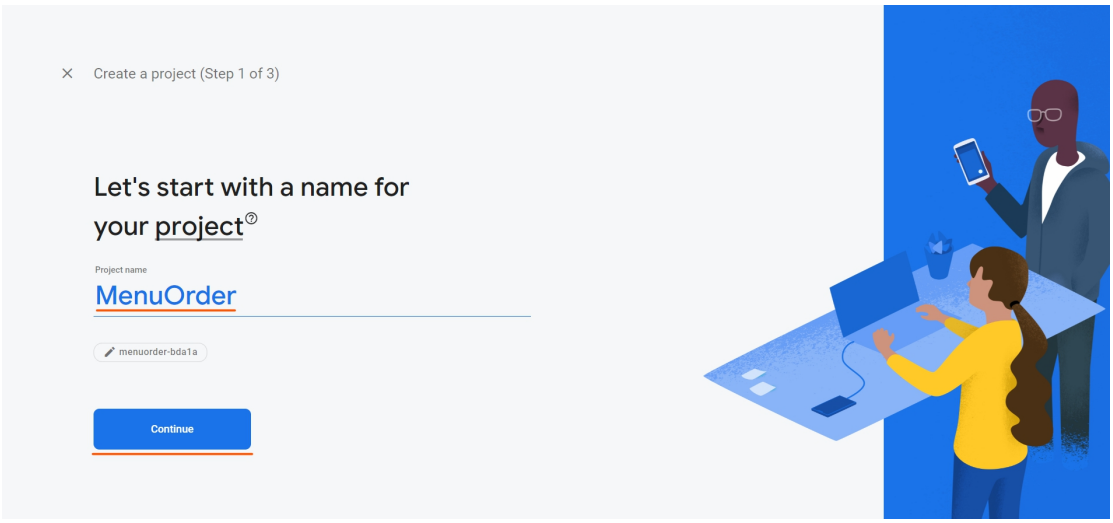
After unzipping, inside application folder run command "npm install" - to get node_modules

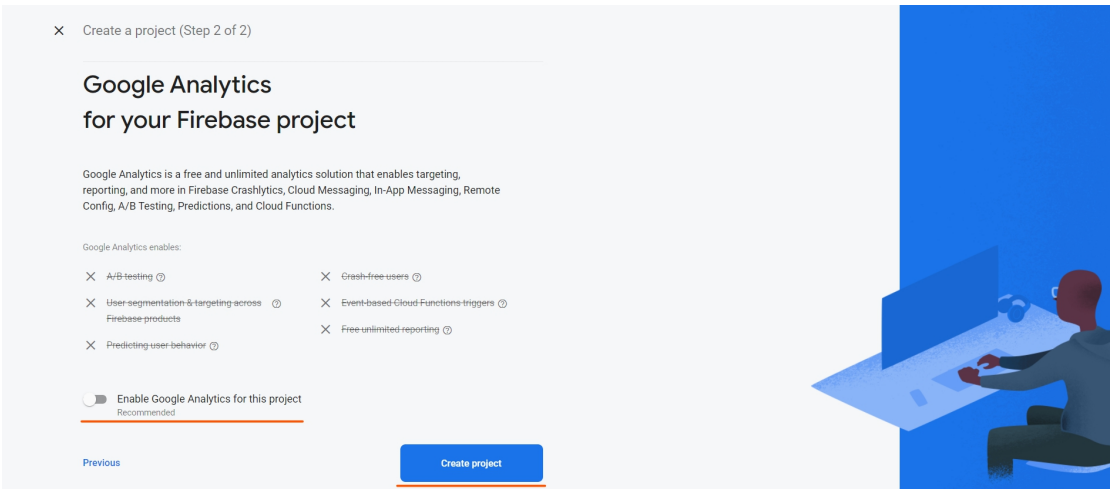
In original project used json-server and axios

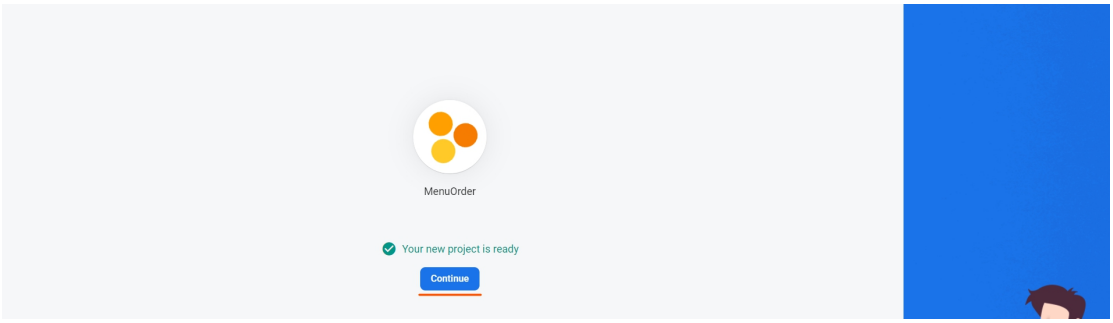
For testing original project (make sure that http://localhost:3000 is free
and json-server already was installed with command "npm install -g json-server")
go to src folder and run command "json-server --watch db.json".
After that, add new Git Bush Terminal and inside main folder of project run command "npm start"

1 Firebase / Create new project (https://console.firebase.google.com/u/0/)

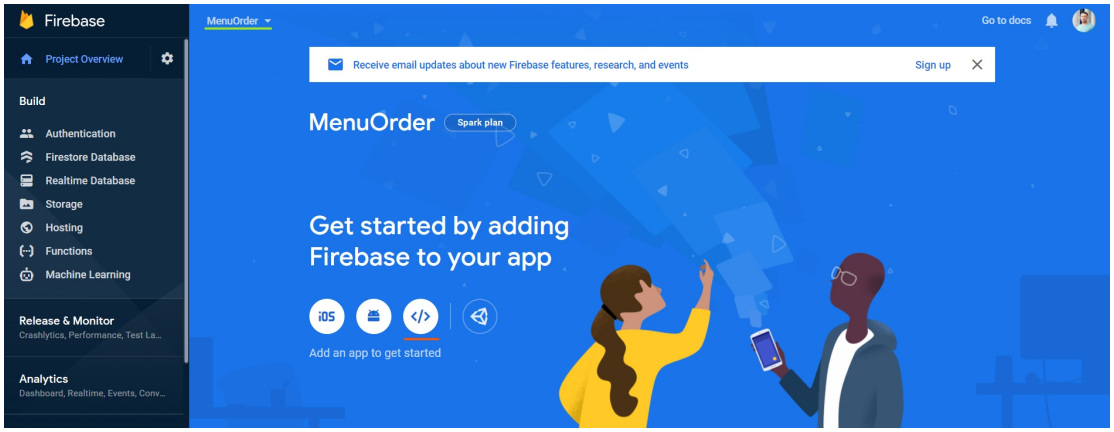


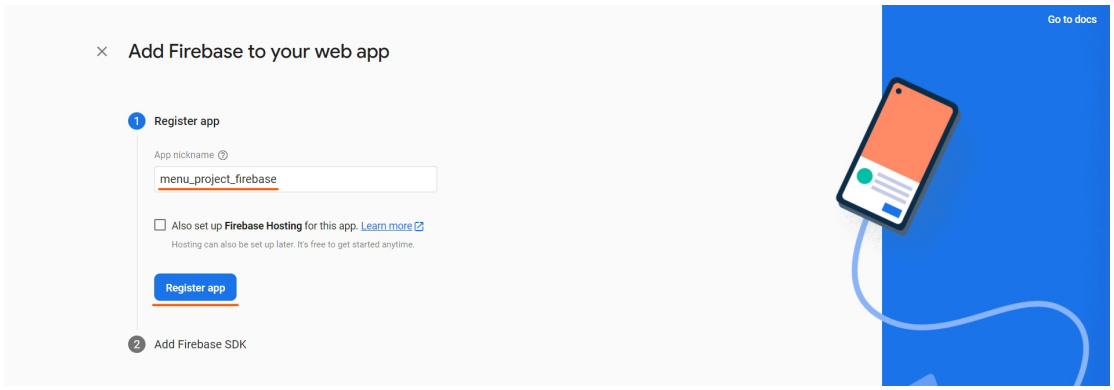


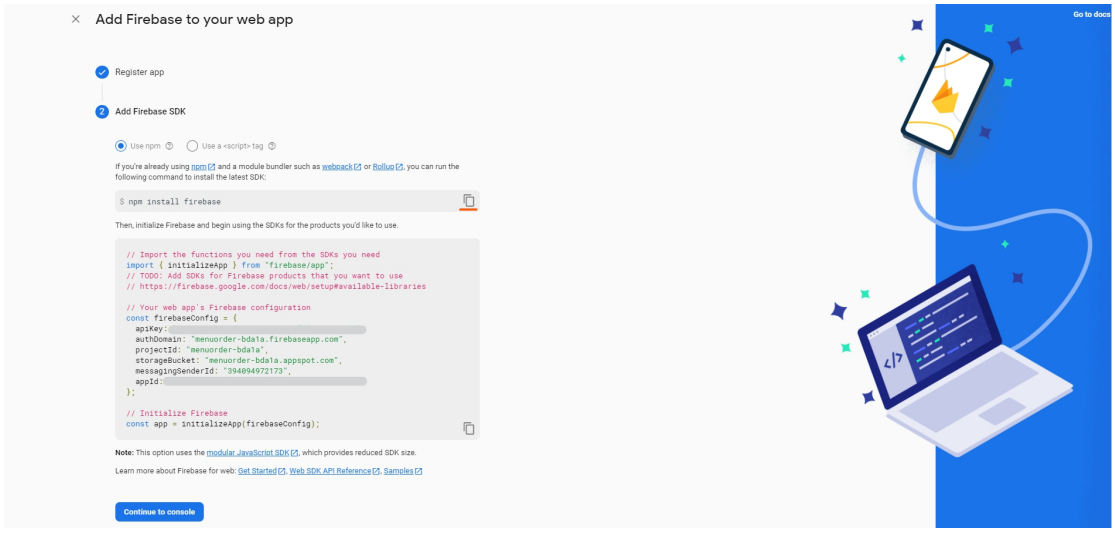


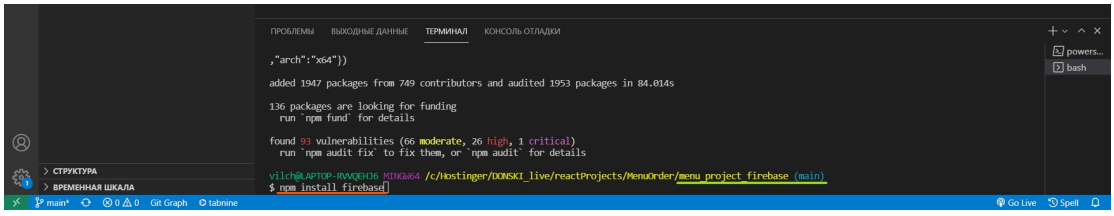


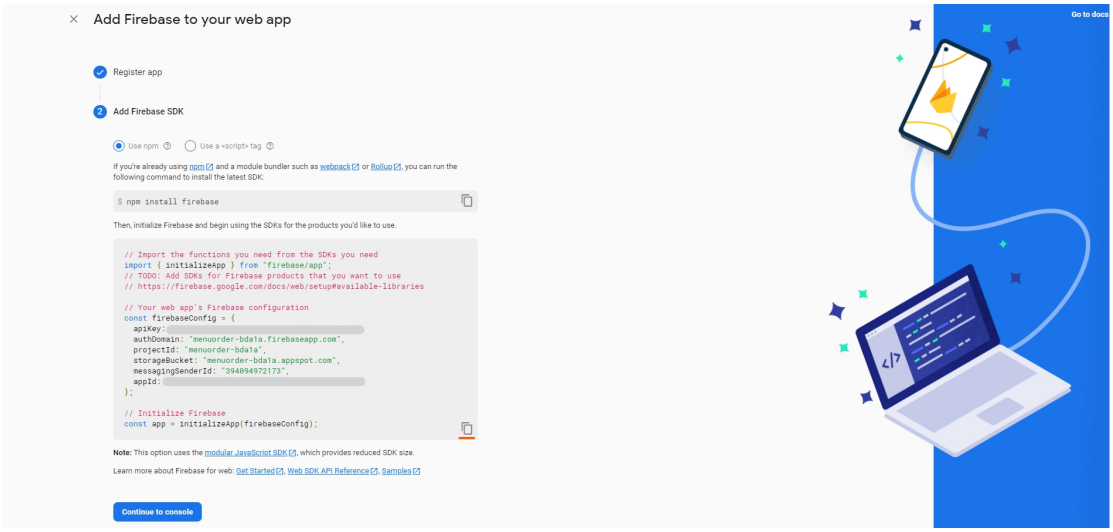
2 Firebase and Visual Studio / Adding Firebase to your App

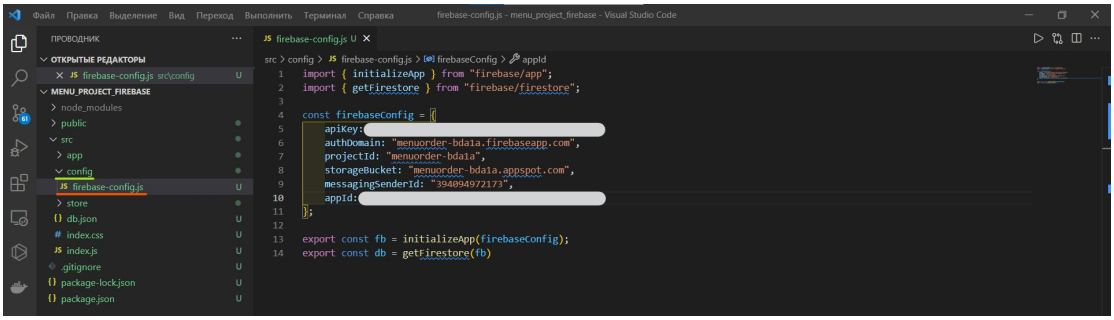


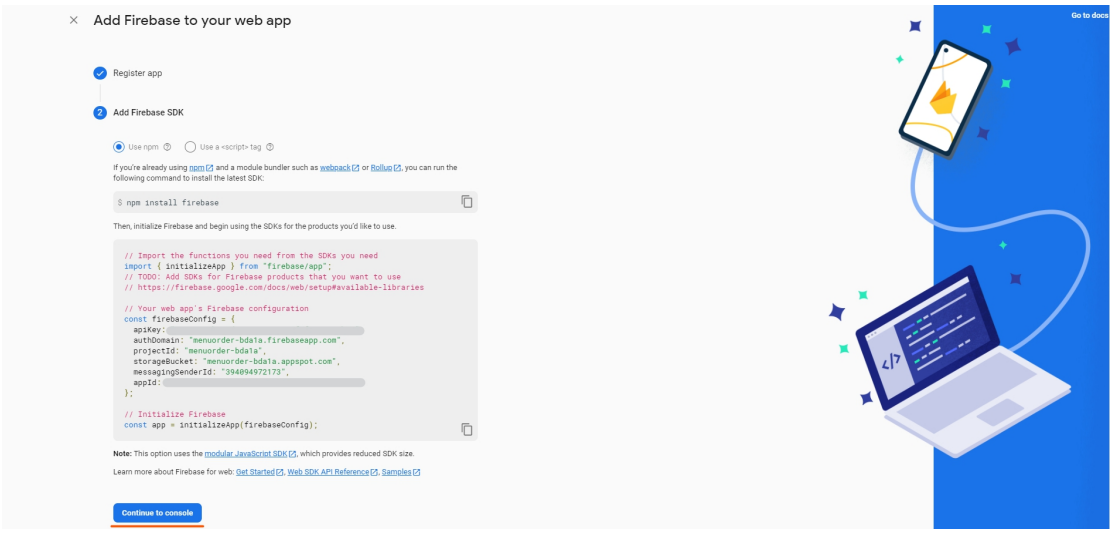




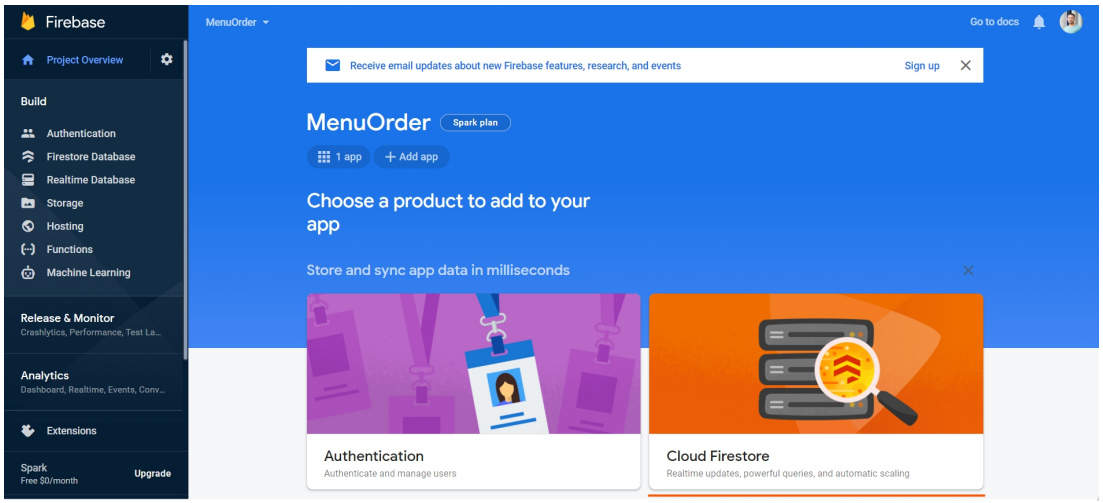


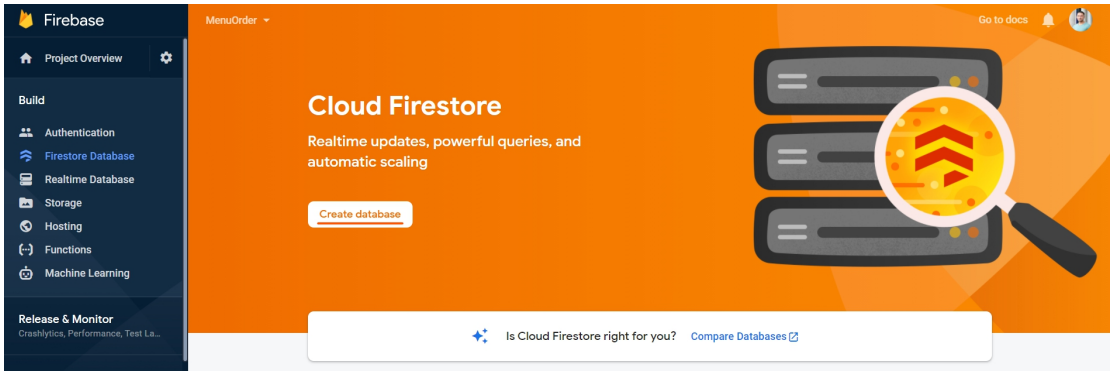

Add Firebase to your web app
Register app
Add Firebase SDK
Use npm
Use a <script> tag
If you're already using npm and a module bundler such as webpack or Rollup, you can run the following command to install the latest SDK:
$ npm install firebase
Then, initialize Firebase and begin using the SDKs for the products you'd like to use.
// Import the functions you need from the SDKs you need
import { initializeApp } from "firebase/app";
// TODO: Add SDKs for Firebase products that you want to use
// https://firebase.google.com/docs/web/setup#available-libraries
// Your web app's Firebase configuration
const firebaseConfig = {
apiKey:
authDomain: "menuorder-bda1a.firebaseapp.com",
projectId: "menuorder-bda1a",
storageBucket: "menuorder-bda1a.appspot.com",
messagingSenderId: "394094972173",
appId:
};
// Initialize Firebase
const app = initializeApp(firebaseConfig);
Note: This option uses the modular JavaScript SDK, which provides reduced SDK size.
Learn more about Firebase for web: Get Started, Web SDK API Reference, Samples
Continue to console
Go to docs

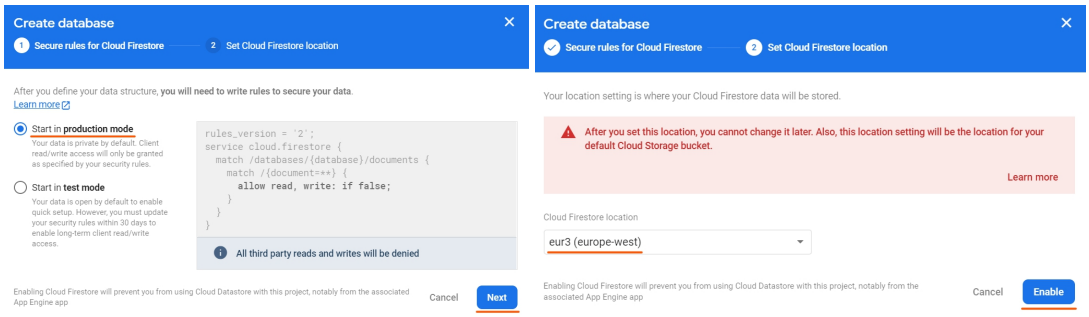

Файл Правка Выделение Вид Переход Выполнить Терминал Справка
firebase-config.js - menu_project_firebase - Visual Studio Code
ПРОВОДНИК
ОТКРЫТЫЕ РЕДАКТОРЫ
firebase-config.js src\config
MENU_PROJECT_FIREBASE
node_modules
public
src
app
config
firebase-config.js
store
db.json
index.css
index.js
.gitignore
package-lock.json
package.json
src > config > firebase-config.js > firebaseConfig > appId
import { initializeApp } from "firebase/app";
import { getFirestore } from "firebase/firestore";
const firebaseConfig = {
apiKey:
authDomain: "menuorder-bda1a.firebaseapp.com",
projectId: "menuorder-bda1a",
storageBucket: "menuorder-bda1a.appspot.com",
messagingSenderId: "394094972173",
appId:
};
export const fb = initializeApp(firebaseConfig);
export const db = getFirestore(fb)


Add Firebase to your web app
Register app
Add Firebase SDK
Use npm
Use a <script> tag
If you're already using npm and a module bundler such as webpack or Rollup, you can run the following command to install the latest SDK:
$ npm install firebase
Then, initialize Firebase and begin using the SDKs for the products you'd like to use.
// Import the functions you need from the SDKs you need
import { initializeApp } from "firebase/app";
// TODO: Add SDKs for Firebase products that you want to use
// https://firebase.google.com/docs/web/setup#available-libraries
// Your web app's Firebase configuration
const firebaseConfig = {
apiKey:
authDomain: "menuorder-bda1a.firebaseapp.com",
projectId: "menuorder-bda1a",
storageBucket: "menuorder-bda1a.appspot.com",
messagingSenderId: "394094972173",
appId:
};
// Initialize Firebase
const app = initializeApp(firebaseConfig);
Note: This option uses the modular JavaScript SDK, which provides reduced SDK size.
Learn more about Firebase for web: Get Started, Web SDK API Reference, Samples
Continue to console
Go to docs

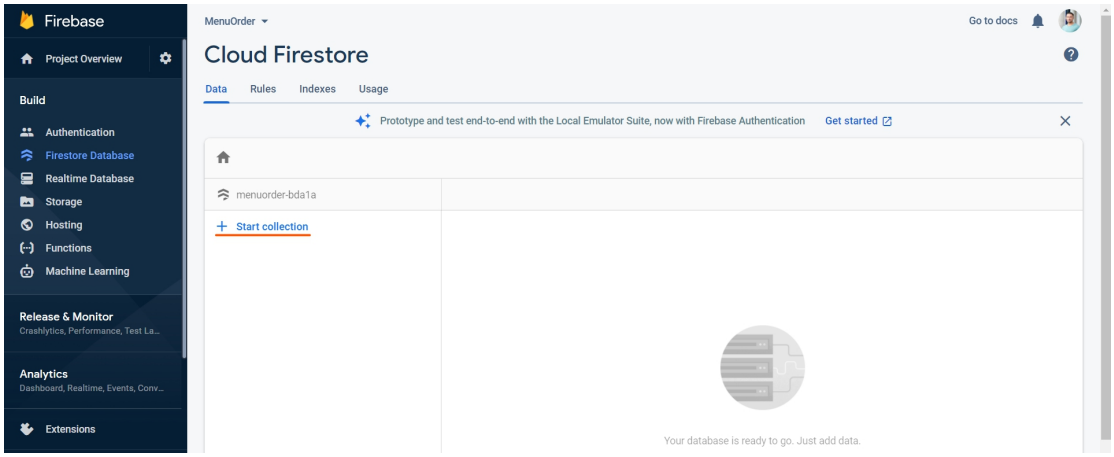
3 Firebase / Create Data-Base

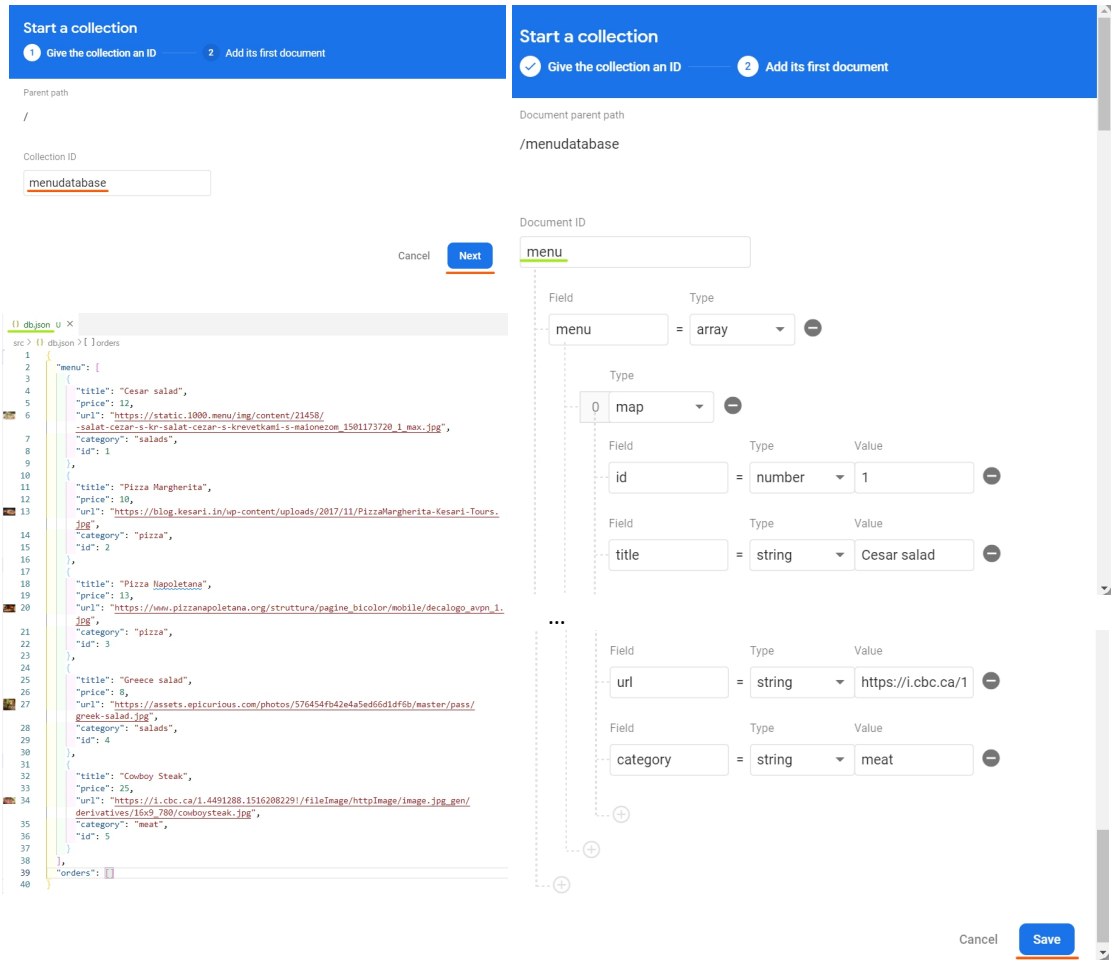


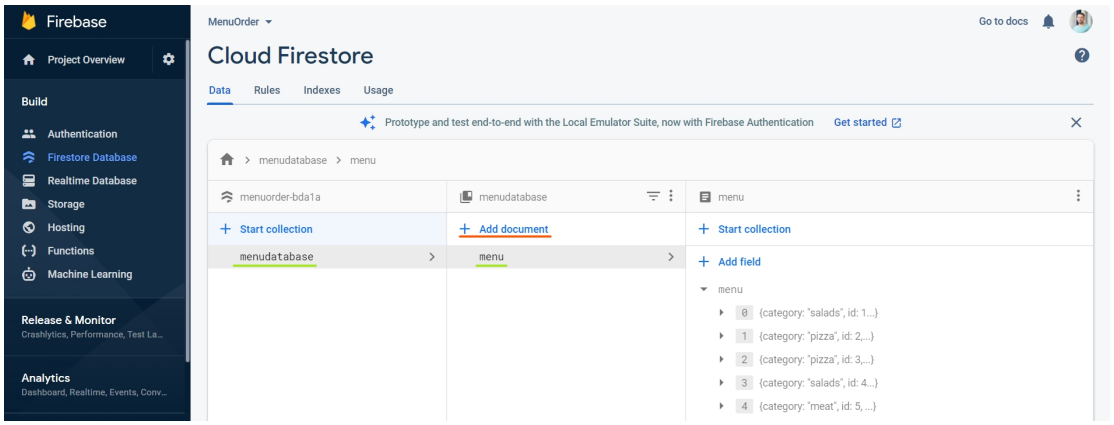


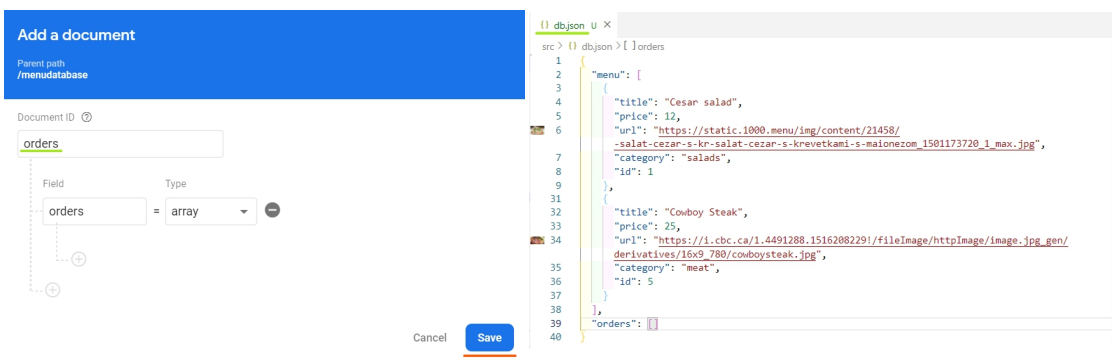


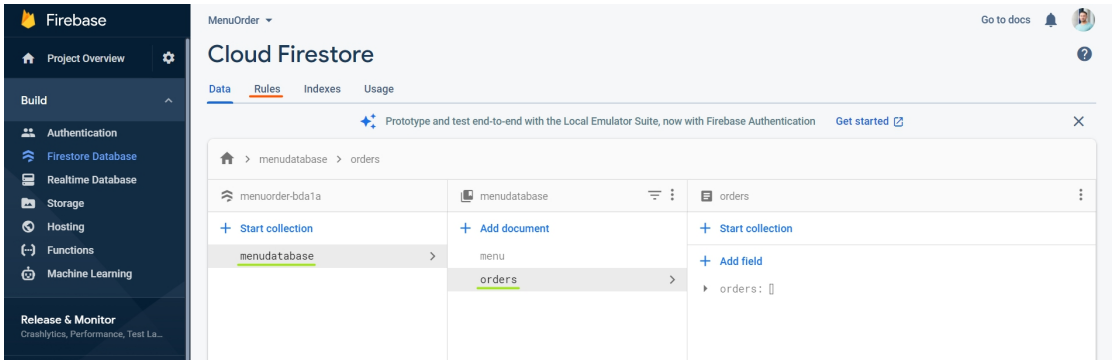
3.1 Firebase / Create collection in Cloud Firestore



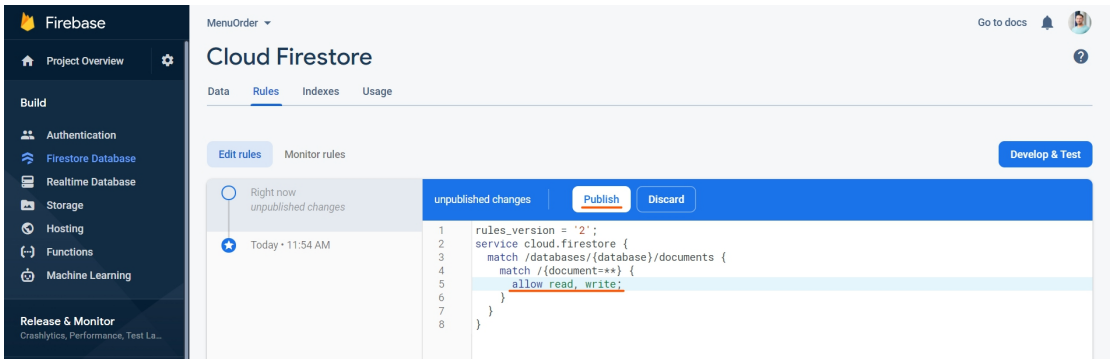

Start a collection
1 Give the collection an ID
2 Add its first document
Parent path
/
Collection ID
menudatabase
Cancel
Next
Start a collection
Give the collection an ID
2 Add its first document
Document parent path
/menudatabase
Document ID
menu
Field
Type
menu
array
Type
0
map
Field
Type
Value
id
number
1
Field
Type
Value
title
string
Cesar salad
Field
Type
Value
url
string
https://i.cbc.ca/1
Field
Type
Value
category
string
meat
Cancel
Save

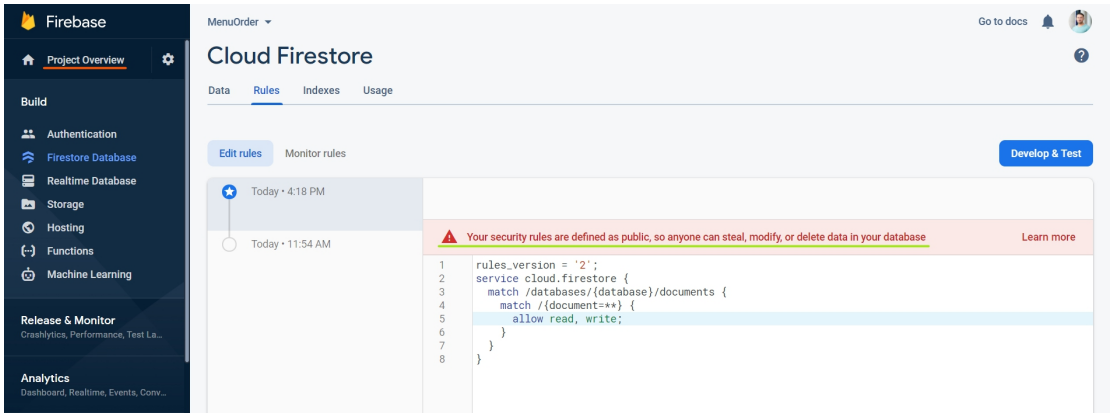

Firebase
Project Overview
Build
Authentication
Firestore Database
Realtime Database
Storage
Hosting
Functions
Machine Learning
Release & Monitor
Analytics
MenuOrder
Go to docs
Cloud Firestore
Data
Rules
Indexes
Usage
Prototype and test end-to-end with the Local Emulator Suite, now with Firebase Authentication
Get started
menudatabase
menu
menuorder-bda1a
Start collection
menudatabase
Add document
menu
Start collection
Add field
menu
0 {category: "salads", id: 1...}
1 {category: "pizza", id: 2,...}
2 {category: "pizza", id: 3,...}
3 {category: "salads", id: 4...}
4 {category: "meat", id: 5, ...}


Add a document
Parent path
/menudatabase
Document ID
orders
Field
Type
orders
array
Cancel
Save
db.json
src > db.json > [ ] orders

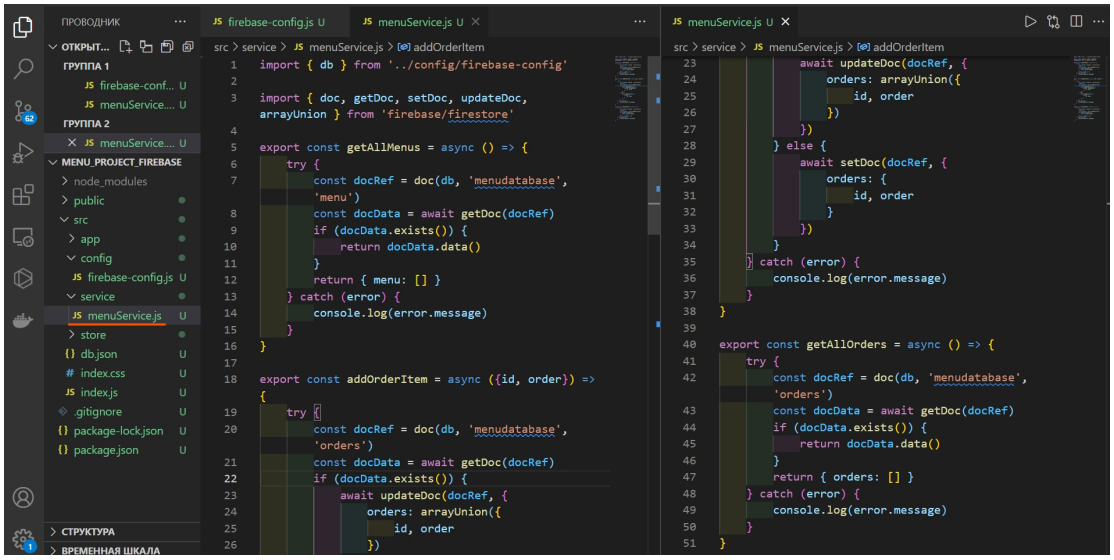


3.2 Firebase / Check (and if needed modify) Rules for your Cloud Firestore



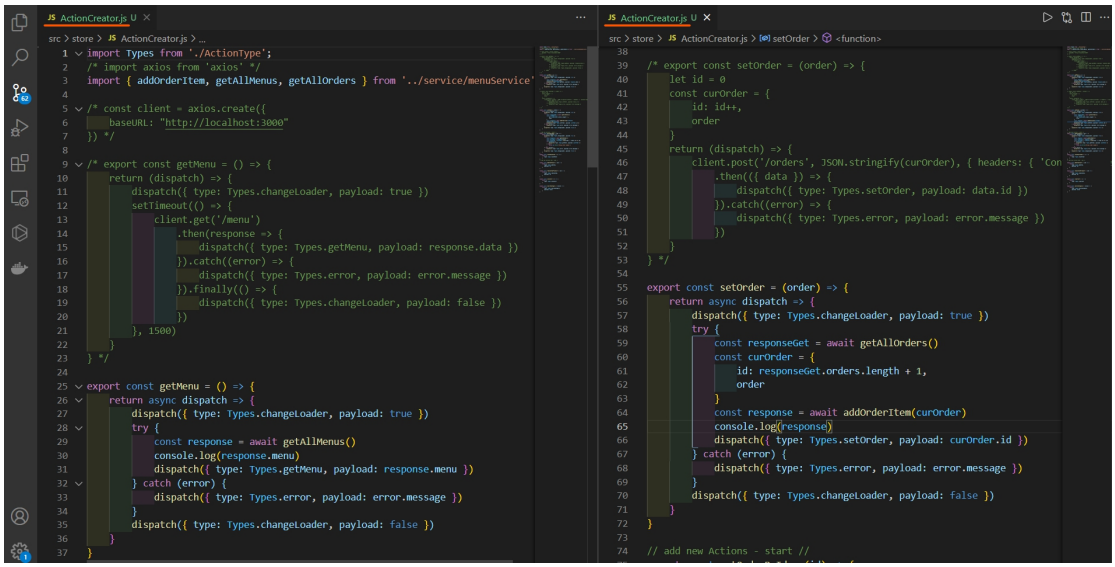


4 Visual Studio / create Functions for communication with Firebase



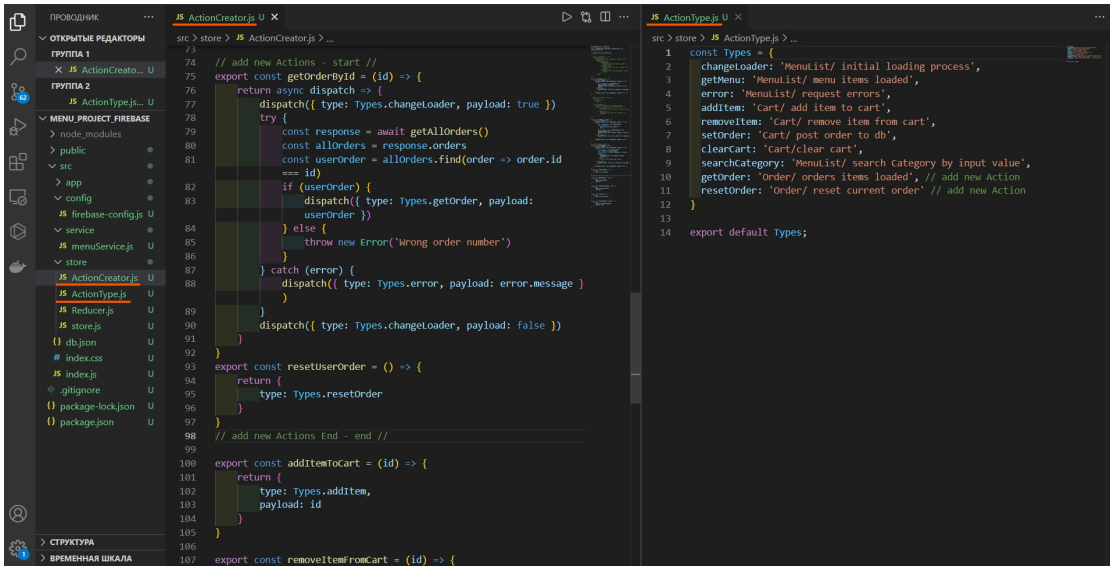
```
import { db } from '../config/firebase-config'

import { doc, getDoc, setDoc, updateDoc,
arrayUnion } from 'firebase/firestore'

export const getAllMenus = async () => {
    try {
        const docRef = doc(db, 'menudatabase',
        'menu')
        const docData = await getDoc(docRef)
        if (docData.exists()) {
            return docData.data()
        }
        return { menu: [] }
    } catch (error) {
        console.log(error.message)
    }
}

export const addOrderItem = async ({id, order}) =>
{
    try {
        const docRef = doc(db, 'menudatabase',
        'orders')
        const docData = await getDoc(docRef)
        if (docData.exists()) {
            await updateDoc(docRef, {
                orders: arrayUnion({
                    id, order
                })
            })
        } else {
            await setDoc(docRef, {
                orders: {
                    id, order
                }
            })
        }
    } catch (error) {
        console.log(error.message)
    }
}

export const getAllOrders = async () => {
    try {
        const docRef = doc(db, 'menudatabase',
        'orders')
        const docData = await getDoc(docRef)
        if (docData.exists()) {
            return docData.data()
        }
        return { orders: [] }
    } catch (error) {
        console.log(error.message)
    }
}
```

5 Visual Studio / modify actions and reducer for communication with Firebase



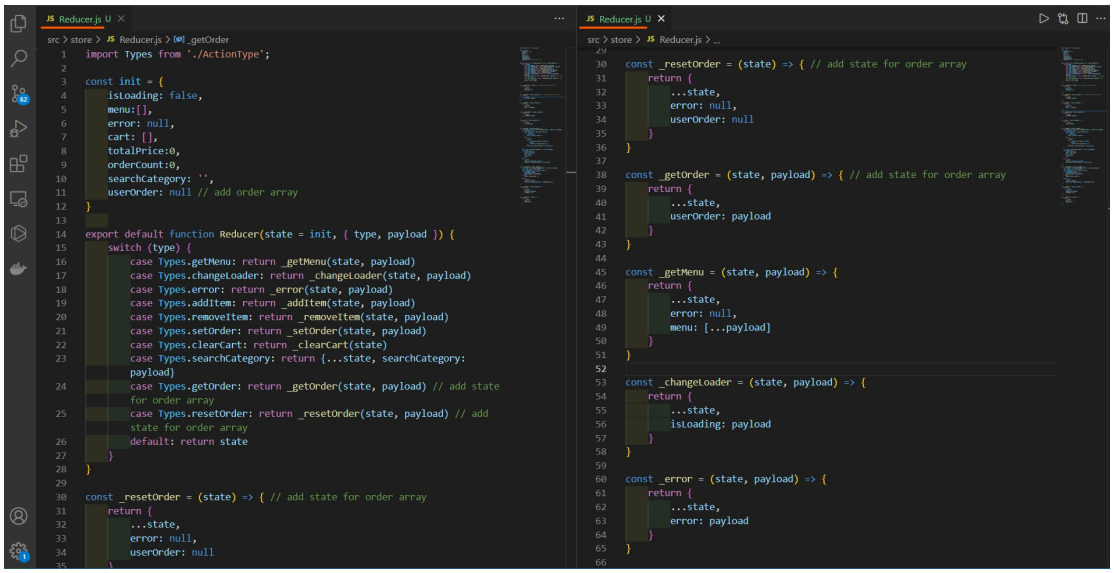
```
import Types from './ActionType';
/* import axios from 'axios' */
import { addOrderItem, getAllMenus, getAllOrders } from '../service/menuService'

/* const client = axios.create({
    baseURL: "http://localhost:3000"
}) */

/* export const getMenu = () => {
    return (dispatch) => {
        dispatch({ type: Types.changeLoader, payload: true })
        setTimeout(() => {
            client.get('/menu')
                .then(response => {
                dispatch({ type: Types.getMenu, payload: response.data })
                }).catch((error) => {
                dispatch({ type: Types.error, payload: error.message })
                }).finally(() => {
                dispatch({ type: Types.changeLoader, payload: false })
                })
        }, 1500)
    }
} */

export const getMenu = () => {
    return async dispatch => {
        dispatch({ type: Types.changeLoader, payload: true })
        try {
            const response = await getAllMenus()
            console.log(response.menu)
            dispatch({ type: Types.getMenu, payload: response.menu })
        } catch (error) {
            dispatch({ type: Types.error, payload: error.message })
        }
        dispatch({ type: Types.changeLoader, payload: false })
    }
}
```

```
/* export const setOrder = (order) => {
    let id = 0
    const curOrder = {
        id: id++,
        order
    }
    return (dispatch) => {
        client.post('/orders', JSON.stringify(curOrder), { headers: { 'Con
            .then(({ data }) => {
                dispatch({ type: Types.setOrder, payload: data.id })
            }).catch((error) => {
                dispatch({ type: Types.error, payload: error.message })
            })
    }
} */

export const setOrder = (order) => {
    return async dispatch => {
        dispatch({ type: Types.changeLoader, payload: true })
        try {
            const responseGet = await getAllOrders()
            const curOrder = {
                id: responseGet.orders.length + 1,
                order
            }
            const response = await addOrderItem(curOrder)
            console.log(response)
            dispatch({ type: Types.setOrder, payload: curOrder.id })
        } catch (error) {
            dispatch({ type: Types.error, payload: error.message })
        }
        dispatch({ type: Types.changeLoader, payload: false })
    }
}

// add new Actions - start //
```

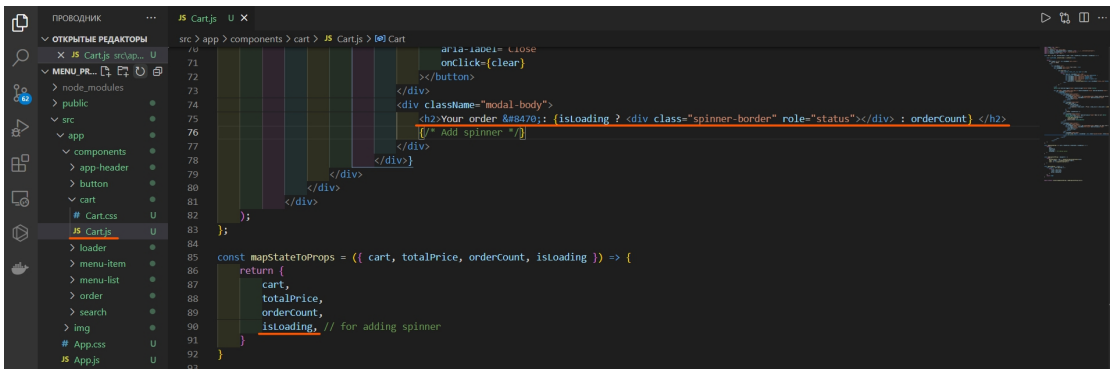


```
// add new Actions - start //
export const getOrderById = (id) => {
    return async dispatch => {
        dispatch({ type: Types.changeLoader, payload: true })
        try {
            const response = await getAllOrders()
            const allOrders = response.orders
            const userOrder = allOrders.find(order => order.id
            === id)
            if (userOrder) {
                dispatch({ type: Types.getOrder, payload:
                userOrder })
            } else {
                throw new Error('Wrong order number')
            }
        } catch (error) {
            dispatch({ type: Types.error, payload: error.message }
            )
        }
        dispatch({ type: Types.changeLoader, payload: false })
    }
}
export const resetUserOrder = () => {
    return {
        type: Types.resetOrder
    }
}
// add new Actions End - end //

export const addItemToCart = (id) => {
    return {
        type: Types.addItem,
        payload: id
    }
}

export const removeItemFromCart = (id) => {
```

```
const Types = {
    changeLoader: 'MenuList/ initial loading process',
    getMenu: 'MenuList/ menu items loaded',
    error: 'MenuList/ request errors',
    addItem: 'Cart/ add item to cart',
    removeItem: 'Cart/ remove item from cart',
    setOrder: 'Cart/ post order to db',
    clearCart: 'Cart/clear cart',
    searchCategory: 'MenuList/ search Category by input value',
    getOrder: 'Order/ orders items loaded', // add new Action
    resetOrder: 'Order/ reset current order' // add new Action
}

export default Types;
```

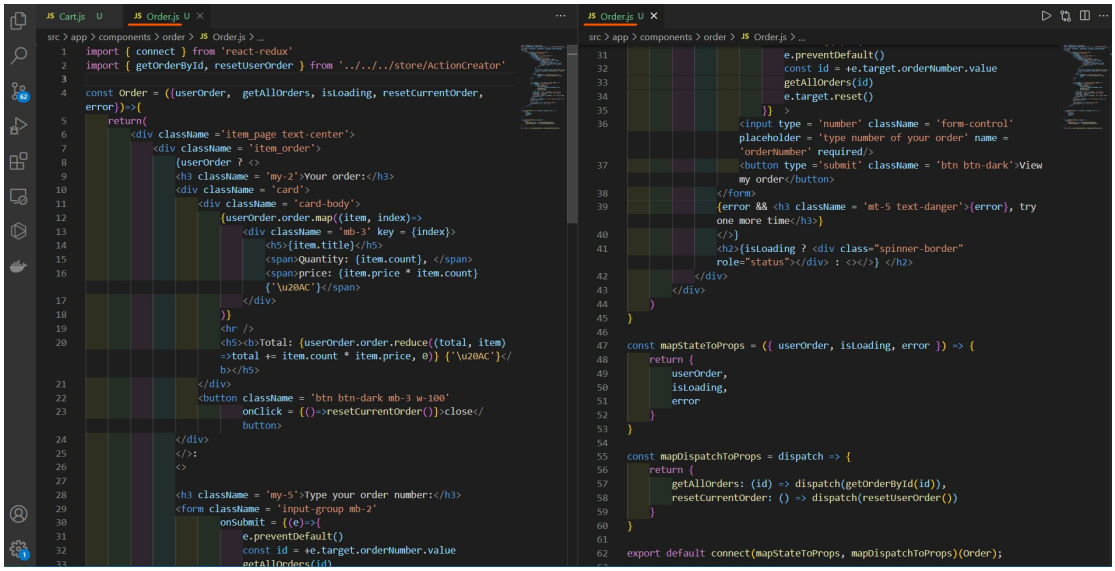


```
import Types from './ActionType';

const init = {
    isLoading: false,
    menu:[],
    error: null,
    cart: [],
    totalPrice:0,
    orderCount:0,
    searchCategory: '',
    userOrder: null // add order array
}

export default function Reducer(state = init, { type, payload }) {
    switch (type) {
        case Types.getMenu: return _getMenu(state, payload)
        case Types.changeLoader: return _changeLoader(state, payload)
        case Types.error: return _error(state, payload)
        case Types.addItem: return _addItem(state, payload)
        case Types.removeItem: return _removeItem(state, payload)
        case Types.setOrder: return _setOrder(state, payload)
        case Types.clearCart: return _clearCart(state)
        case Types.searchCategory: return {...state, searchCategory:
        payload}
        case Types.getOrder: return _getOrder(state, payload) // add state
        for order array
        case Types.resetOrder: return _resetOrder(state, payload) // add
        state for order array
        default: return state
    }
}

const _resetOrder = (state) => { // add state for order array
    return {
        ...state,
        error: null,
        userOrder: null
    }
}

const _getOrder = (state, payload) => { // add state for order array
    return {
        ...state,
        userOrder: payload
    }
}

const _getMenu = (state, payload) => {
    return {
        ...state,
        error: null,
        menu: [...payload]
    }
}

const _changeLoader = (state, payload) => {
    return {
        ...state,
        isLoading: payload
    }
}

const _error = (state, payload) => {
    return {
        ...state,
        error: payload
    }
}
```



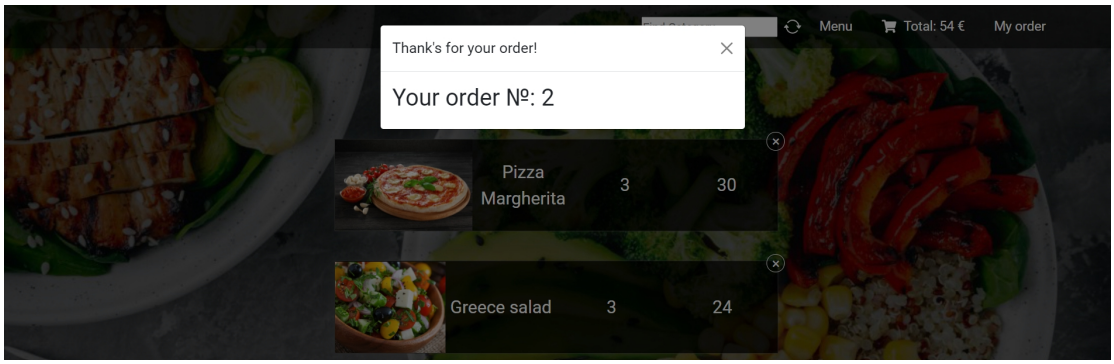
```
                                onClick={clear}
                            ></button>
                        </div>
                        <div className="modal-body">
                            <h2>Your order &#8470;: {isLoading ? <div class="spinner-border" role="status"></div> : orderCount} </h2>
                            {/* Add spinner */}
                        </div>
                    </div>}
                </div>
            </div>
        </div>
    );
};

const mapStateToProps = ({ cart, totalPrice, orderCount, isLoading }) => {
    return {
        cart,
        totalPrice,
        orderCount,
        isLoading, // for adding spinner
    }
}
```

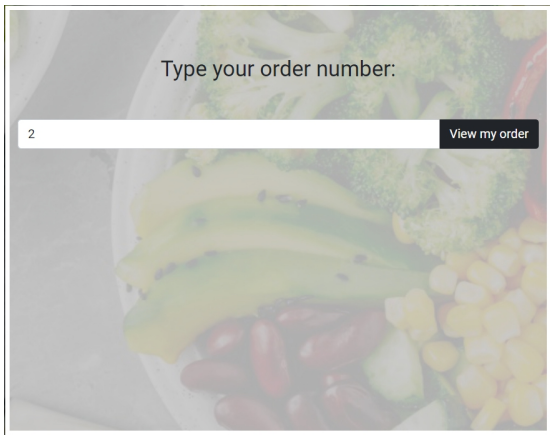
6 Visual Studio / modify your application for new actions

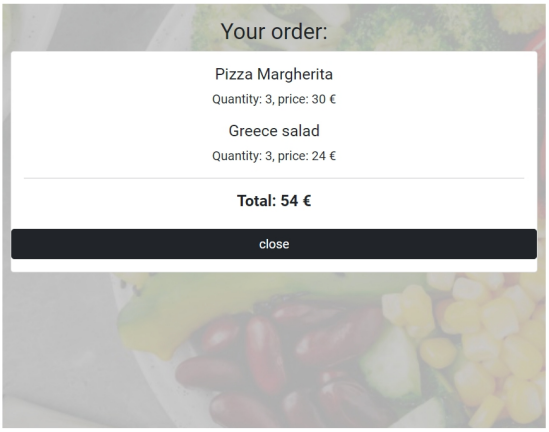


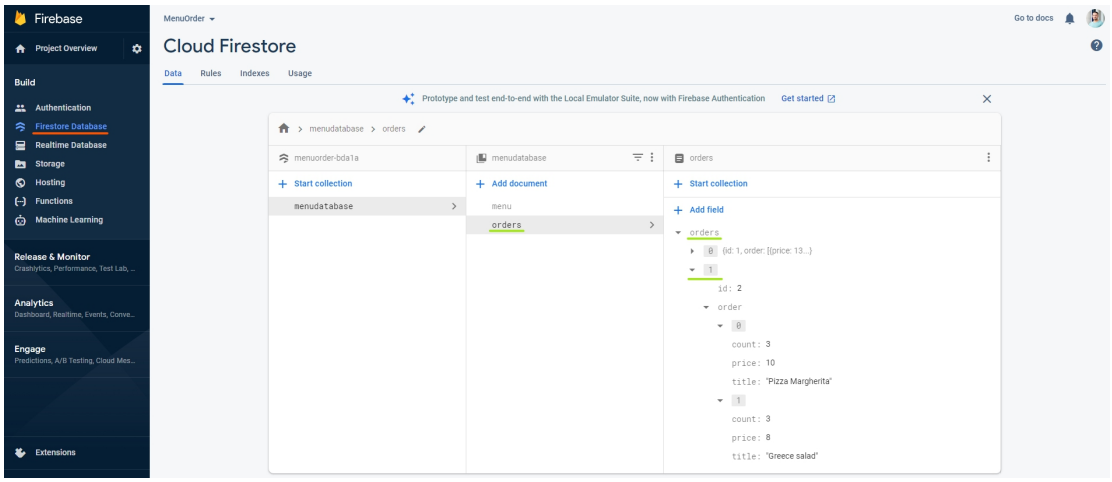
```
import { connect } from 'react-redux'
import { getOrderById, resetUserOrder } from '../../../store/ActionCreator'

const Order = ({userOrder,  getAllOrders, isLoading, resetCurrentOrder,
error})=>{
    return(
        <div className ='item_page text-center'>
            <div className = 'item_order'>
                {userOrder ? <>
                <h3 className = 'my-2'>Your order:</h3>
                <div className = 'card'>
                    <div className = 'card-body'>
                        {userOrder.order.map((item, index)=>
                            <div className = 'mb-3' key = {index}>
                                <h5>{item.title}</h5>
                                <span>Quantity: {item.count}, </span>
                                <span>price: {item.price * item.count}
                                {'\u20AC'}</span>
                            </div>
                        )}
                        <hr />
                        <h5><b>Total: {userOrder.order.reduce((total, item)
                        =>total += item.count * item.price, 0)} {'\u20AC'}</
                        b></h5>
                    </div>
                    <button className = 'btn btn-dark mb-3 w-100'
                            onClick = {()=>resetCurrentOrder()}>close</
                            button>
                </div>
                </>:
                <>

                <h3 className = 'my-5'>Type your order number:</h3>
                <form className = 'input-group mb-2'
                        onSubmit = {(e)=>{
                            e.preventDefault()
                            const id = +e.target.orderNumber.value
                            getAllOrders(id)
                            e.target.reset()
                        }} >
                    <input type = 'number' className = 'form-control'
                    placeholder = 'type number of your order' name =
                    'orderNumber' required/>
                    <button type ='submit' className = 'btn btn-dark'>View
                    my order</button>
                </form>
                {error && <h3 className = 'mt-5 text-danger'>{error}, try
                one more time</h3>}
                </>}
                <h2>{isLoading ? <div class="spinner-border"
                role="status"></div> : <></>} </h2>
            </div>
        </div>
    )
}

const mapStateToProps = ({ userOrder, isLoading, error }) => {
    return {
        userOrder,
        isLoading,
        error
    }
}

const mapDispatchToProps = dispatch => {
    return {
        getAllOrders: (id) => dispatch(getOrderById(id)),
        resetCurrentOrder: () => dispatch(resetUserOrder())
    }
}

export default connect(mapStateToProps, mapDispatchToProps)(Order);
```

7 Visual Studio and Firebase / test your modified application and Cloud Firestore

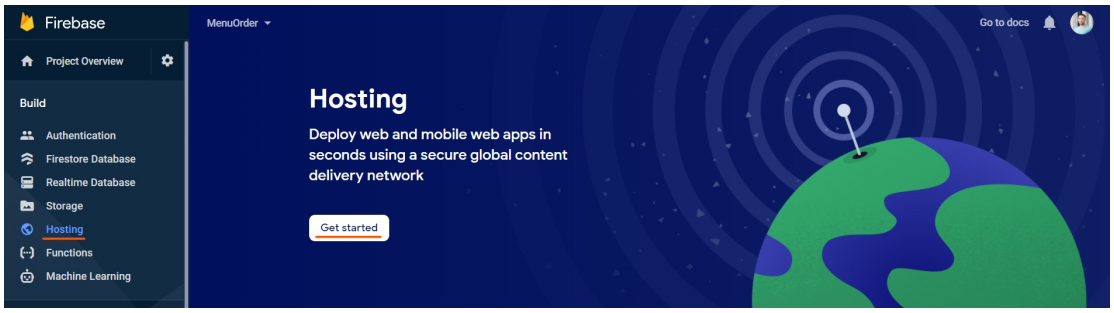


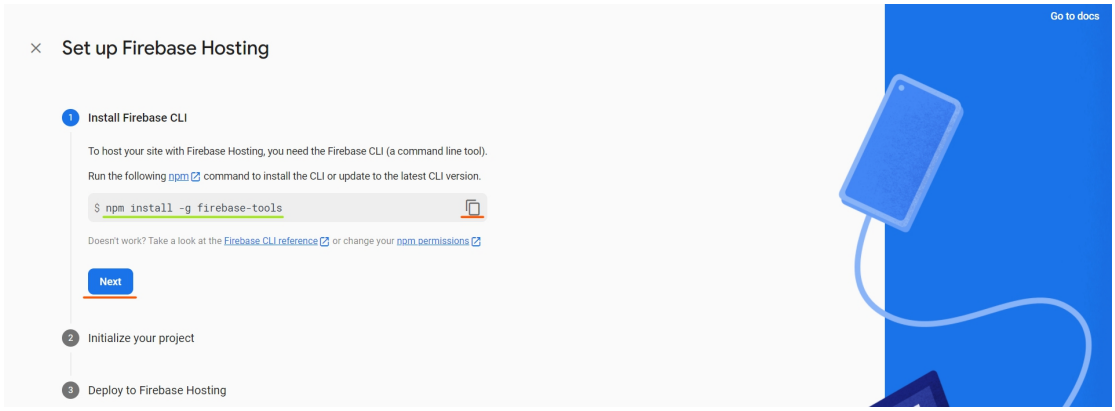


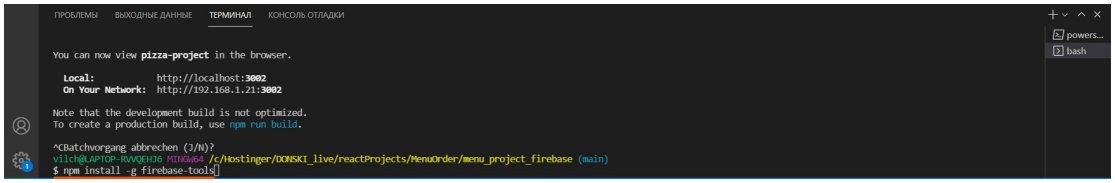


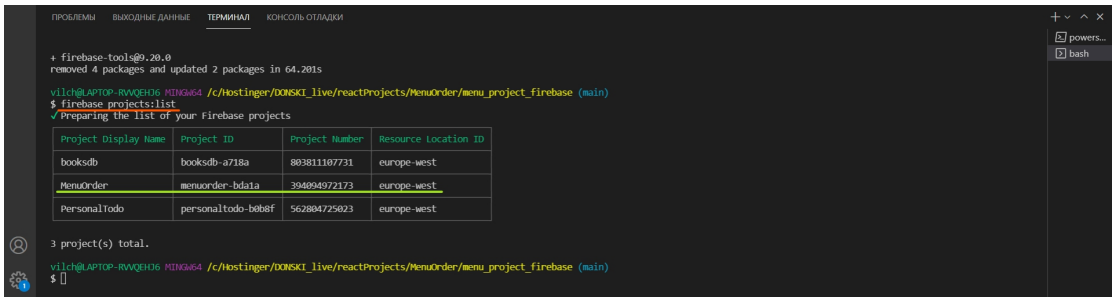


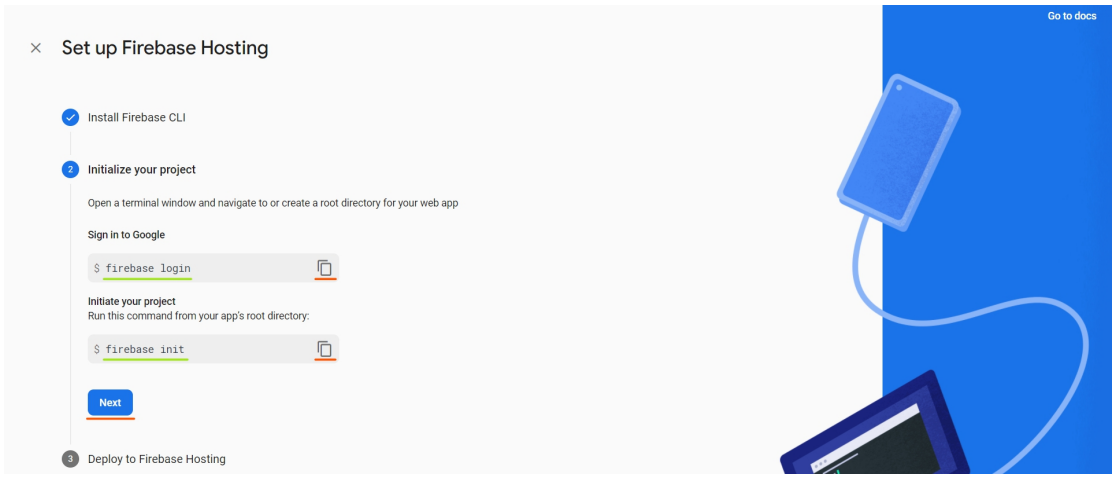
8 Firebase and Visual Studio / create Hosting for your application

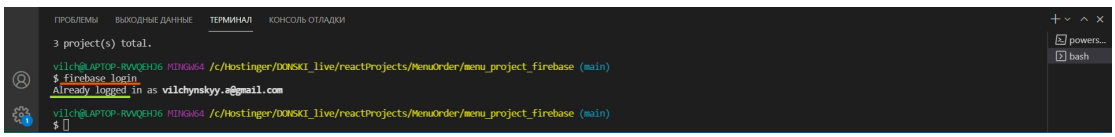




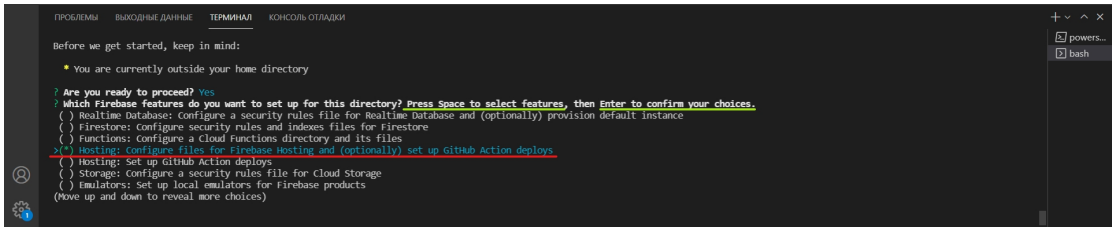




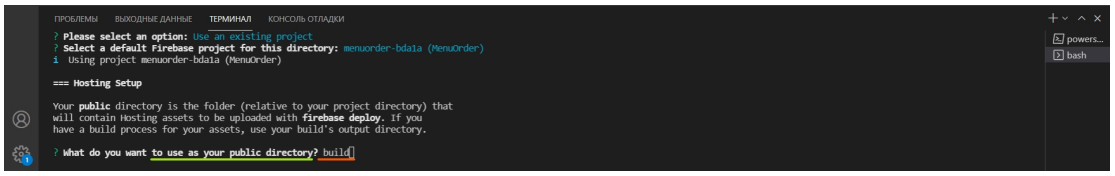





ПРОБЛЕМЫ ВЫХОДНЫЕ ДАННЫЕ ТЕРМИНАЛ КОНСОЛЬ ОТЛАДКИ
powers...
bash
vilch@LAPTOP-RVVQEHJ6 MINGW64 /c/Hostinger/DONSKI_live/reactProjects/MenuOrder/menu_project_firebase (main)
$ firebase login
Already logged in as vilchynskyy.a@gmail.com
vilch@LAPTOP-RVVQEHJ6 MINGW64 /c/Hostinger/DONSKI_live/reactProjects/MenuOrder/menu_project_firebase (main)
$ firebase init
You're about to initialize a Firebase project in this directory:
C:\Hostinger\DONSKI_live\reactProjects\MenuOrder\menu_project_firebase
Before we get started, keep in mind:
* You are currently outside your home directory
? Are you ready to proceed? (Y/n) y


ПРОБЛЕМЫ ВЫХОДНЫЕ ДАННЫЕ ТЕРМИНАЛ КОНСОЛЬ ОТЛАДКИ
powers...
bash
Before we get started, keep in mind:
* You are currently outside your home directory
? Are you ready to proceed? Yes
? Which Firebase features do you want to set up for this directory? Press Space to select features, then Enter to confirm your choices.
( ) Realtime Database: Configure a security rules file for Realtime Database and (optionally) provision default instance
( ) Firestore: Configure security rules and indexes files for Firestore
( ) Functions: Configure a Cloud Functions directory and its files
>(*) Hosting: Configure files for Firebase Hosting and (optionally) set up GitHub Action deploys
( ) Hosting: Set up GitHub Action deploys
( ) Storage: Configure a security rules file for Cloud Storage
( ) Emulators: Set up local emulators for Firebase products
(Move up and down to reveal more choices)

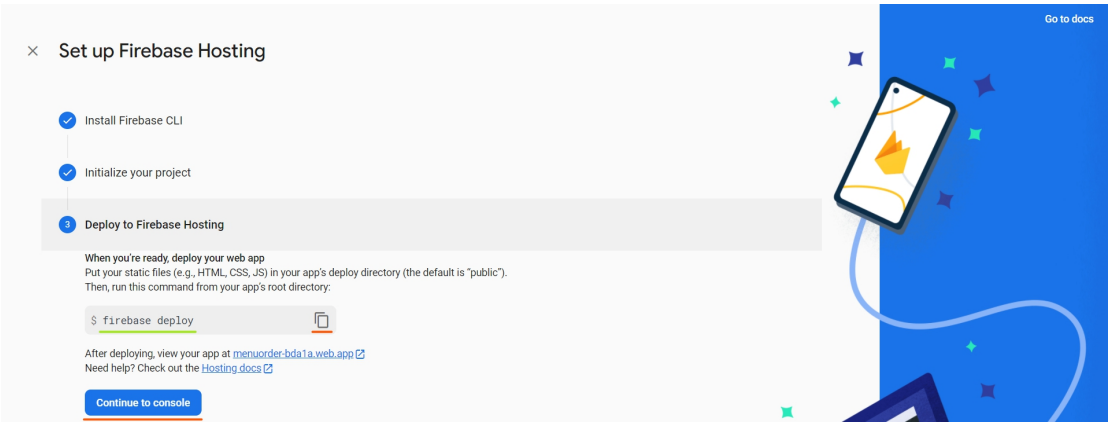

ПРОБЛЕМЫ ВЫХОДНЫЕ ДАННЫЕ ТЕРМИНАЛ КОНСОЛЬ ОТЛАДКИ
powers...
bash
? Are you ready to proceed? Yes
? Which Firebase features do you want to set up for this directory? Press Space to select features, then Enter to confirm your choices. Hosting: Configure files for Firebase Hosting and (optionally) set up GitHub Action deploys
=== Project Setup
First, let's associate this project directory with a Firebase project.
You can create multiple project aliases by running firebase use --add,
but for now we'll just set up a default project.
? Please select an option: (Use arrow keys)
> Use an existing project
Create a new project
Add Firebase to an existing Google Cloud Platform project
Don't set up a default project


ПРОБЛЕМЫ ВЫХОДНЫЕ ДАННЫЕ ТЕРМИНАЛ КОНСОЛЬ ОТЛАДКИ
powers...
bash
? Please select an option: Use an existing project
? Select a default Firebase project for this directory: menuorder-bda1a (MenuOrder)
i Using project menuorder-bda1a (MenuOrder)
=== Hosting Setup
Your public directory is the folder (relative to your project directory) that
will contain Hosting assets to be uploaded with firebase deploy. If you
have a build process for your assets, use your build's output directory.
? What do you want to use as your public directory? build


ПРОБЛЕМЫ ВЫХОДНЫЕ ДАННЫЕ ТЕРМИНАЛ КОНСОЛЬ ОТЛАДКИ
powers...
bash
i Using project menuorder-bda1a (MenuOrder)
=== Hosting Setup
Your public directory is the folder (relative to your project directory) that
will contain Hosting assets to be uploaded with firebase deploy. If you
have a build process for your assets, use your build's output directory.
? What do you want to use as your public directory? build
? Configure as a single-page app (rewrite all urls to /index.html)? (y/N) y


ПРОБЛЕМЫ ВЫХОДНЫЕ ДАННЫЕ ТЕРМИНАЛ КОНСОЛЬ ОТЛАДКИ
powers...
bash
i Using project menuorder-bda1a (MenuOrder)
=== Hosting Setup
Your public directory is the folder (relative to your project directory) that
will contain Hosting assets to be uploaded with firebase deploy. If you
have a build process for your assets, use your build's output directory.
? What do you want to use as your public directory? build
? Configure as a single-page app (rewrite all urls to /index.html)? Yes
? Set up automatic builds and deploys with GitHub? (y/N) N


ПРОБЛЕМЫ ВЫХОДНЫЕ ДАННЫЕ ТЕРМИНАЛ КОНСОЛЬ ОТЛАДКИ
powers...
bash
? What do you want to use as your public directory? build
? Configure as a single-page app (rewrite all urls to /index.html)? Yes
? Set up automatic builds and deploys with GitHub? No
+ Wrote build/index.html
i Writing configuration info to firebase.json...
i Writing project information to .firebaserc...
+ Firebase initialization complete!
vilch@LAPTOP-RVVQEHJ6 MINGW64 /c/Hostinger/DONSKI_live/reactProjects/MenuOrder/menu_project_firebase (main)
$ npm run build



9 Firebase / test your Hosting