Spis treści

[**1.** **RWD** 4](#_Toc507179246)

[1.1. max-device-width ogarnąć 4](#_Toc507179247)

[**2.** **CSS** 5](#_Toc507179248)

[2.1. BEM 5](#_Toc507179249)

[2.2. Kaskadowość 5](#_Toc507179250)

[2.3. Dziedziczenie 5](#_Toc507179251)

[2.4. Specyficzność 5](#_Toc507179252)

[**3.** **Bootstrap** 5](#_Toc507179253)

[3.1. kiedy i dlaczego używać 5](#_Toc507179254)

[**4.** **SASS** 5](#_Toc507179255)

[4.1. po co używać 5](#_Toc507179256)

[4.2. loader do reacta 5](#_Toc507179257)

[4.3. co ma ponad css 5](#_Toc507179258)

[4.4. tworzenie plików css preprocesorem 5](#_Toc507179259)

[4.5. zobaczyć LESS – co to i podstawowe różnice do SASS 5](#_Toc507179260)

[4.6. SCSS 5](#_Toc507179261)

[**5.** **GIT i jego funkcje** 5](#_Toc507179262)

[5.1. Mergowanie 6](#_Toc507179263)

[5.2. Branchowanie 6](#_Toc507179264)

[5.3. pull requesty 6](#_Toc507179265)

[5.4. cofanie się do commitów 6](#_Toc507179266)

[5.5. Stash 6](#_Toc507179267)

[5.6. Konflikty - skąd się biorą i jak je rozwiązywać 6](#_Toc507179268)

[5.7. poprosić Ozza 6](#_Toc507179269)

[5.8. git flow 6](#_Toc507179270)

[**6.** **Npm, a yarn** 6](#_Toc507179271)

[6.1. popytać Ozza 6](#_Toc507179272)

[**7.** **JS** 6](#_Toc507179273)

[7.1. Ify 6](#_Toc507179274)

[7.2. Pętle 6](#_Toc507179275)

[7.3. Funkcje 6](#_Toc507179276)

[7.4. Prototypy 6](#_Toc507179277)

[7.5. operatory logiczne 7](#_Toc507179278)

[7.6. undefined vs null 7](#_Toc507179279)

[7.7. Asynchroniczność 7](#_Toc507179280)

[**8.** **JQuery** 7](#_Toc507179281)

[8.1. Funkcjonalności 7](#_Toc507179282)

[8.2. dlaczego wypadł z łask 7](#_Toc507179283)

[8.3. zalety na swój czas 7](#_Toc507179284)

[**9.** **OOP** 7](#_Toc507179285)

[9.1. dlaczego obiektowość 7](#_Toc507179286)

[9.2. implementacja w react 7](#_Toc507179287)

[**10.** **Ajax** 7](#_Toc507179288)

[10.1. ajax a redux 7](#_Toc507179289)

[10.2. po co używać 7](#_Toc507179290)

[10.3. zady i walety 7](#_Toc507179291)

[**11.** **API** 7](#_Toc507179292)

[11.1. api na stronach 7](#_Toc507179293)

[11.2. api do mongo w mern 8](#_Toc507179294)

[11.3. hash użytkownika API 8](#_Toc507179295)

[**12.** **Node** 8](#_Toc507179296)

[12.1. przypomnieć sobie jak postawić serwer na czystym node 8](#_Toc507179297)

[12.2. różnice z php 8](#_Toc507179298)

[12.3. back-end – serwer i zapytania do niego 8](#_Toc507179299)

[**13.** **ReactJS** 8](#_Toc507179300)

[13.1. szablony (PuG EJS) 8](#_Toc507179301)

[13.2. struktura danych 8](#_Toc507179302)

[13.3. cykl życia komponentu 8](#_Toc507179303)

[13.4. ReactDOM 8](#_Toc507179304)

[13.5. state vs props 8](#_Toc507179305)

[13.6. dlaczego React 8](#_Toc507179306)

[13.7. podstawowe różnice z AgularJS 8](#_Toc507179307)

[**14.** **ES6** 8](#_Toc507179308)

[14.1. co wprowadza 8](#_Toc507179309)

[14.2. jaki sposób kompiluje się do es2015 (babel) 9](#_Toc507179310)

[**15.** **Webpack** 9](#_Toc507179311)

[15.1. Ozz 9](#_Toc507179312)

[15.2. Pluginy 9](#_Toc507179313)

[15.3. Loadery 9](#_Toc507179314)

[15.4. HMR 9](#_Toc507179315)

[**16.** **ExpressJS** 9](#_Toc507179316)

[16.1. Middleware 9](#_Toc507179317)

[16.2. routing – endpointy 9](#_Toc507179318)

[16.3. CRUD 9](#_Toc507179319)

[16.4. JSON w JS 9](#_Toc507179320)

[16.5. autoryzacja 9](#_Toc507179321)

[16.6. serwowanie plików 9](#_Toc507179322)

[**17.** **Inne** 9](#_Toc507179323)

[17.1. Makiety 9](#_Toc507179324)

[17.2. SCRUM 9](#_Toc507179325)

[17.3. Polling 10](#_Toc507179326)

[17.4. Streaming 10](#_Toc507179327)

[17.5. Long-polling 10](#_Toc507179328)

[17.6. Socket.io 10](#_Toc507179329)

[**18.** **Redux** 10](#_Toc507179330)

[18.1. Flux 10](#_Toc507179331)

[18.2. Cqrs 10](#_Toc507179332)

[18.3. event sourcing 10](#_Toc507179333)

[18.4. zasady korzystania 10](#_Toc507179334)

[18.5. programowanie funkcyjne 10](#_Toc507179335)

[18.6. store w mern 10](#_Toc507179336)

[18.7. React-redux 10](#_Toc507179337)

[18.8. middleware redux 10](#_Toc507179338)

[18.9. data flow 10](#_Toc507179339)

[**19.** **React-router** 10](#_Toc507179340)

[19.1. co to jest i jak działa 10](#_Toc507179341)

[**20.** **MongoDB** 11](#_Toc507179342)

[20.1. nierelacyjne bazy danych 11](#_Toc507179343)

[20.2. SQL 11](#_Toc507179344)

[20.3. Mongoose 11](#_Toc507179345)

[20.4. operacje na bazie Mongo 11](#_Toc507179346)

[**21.** **MERN** 11](#_Toc507179347)

[21.1. inne wersje np. MEAN 11](#_Toc507179348)

[21.2. kompletnie inne technologie PERL 11](#_Toc507179349)

[21.3. aplikacje izomorficzne 11](#_Toc507179350)

[21.4. server side rendering 11](#_Toc507179351)

[21.5. wirtualizacja (VBox, vagrant, docker) – Ozz 11](#_Toc507179352)

[21.6. generator kodu 11](#_Toc507179353)

[21.7. Redux-thunk 11](#_Toc507179354)

[21.8. Fetch 11](#_Toc507179355)

[21.9. przejrzeć dokumentację 11](#_Toc507179356)

[**22.** **Pozostałe** 11](#_Toc507179357)

[22.1. REST API 12](#_Toc507179358)

[22.2. normalizacja danych 12](#_Toc507179359)

[22.3. ReactDnD 12](#_Toc507179360)

[22.4. compose i connect 12](#_Toc507179361)

[22.5. obadać punkty połączenia frontu z back-endem 12](#_Toc507179362)

[**23.** **Zasady** 12](#_Toc507179363)

[23.1. DRY 12](#_Toc507179364)

[23.2. KISS 12](#_Toc507179365)

[23.3. poziomy abstrakcji 12](#_Toc507179366)

[23.4. SOLID Principles 12](#_Toc507179367)

[**24.** **Wzorce projektowe** 12](#_Toc507179368)

[**25.** **Migracje** 12](#_Toc507179369)

[**26.** **Zagadnienia serwerowe** 12](#_Toc507179370)

[26.1. deployowanie, CI/CD 12](#_Toc507179371)

[26.2. Cache’owanie (memcache, redis, cloudflare) 12](#_Toc507179372)

[26.3. Domeny (DNS, Vhosty) 12](#_Toc507179373)

1. **RWD**
   1. max-device-width ogarnąć
2. **CSS**
   1. BEM
   2. Kaskadowość
   3. Dziedziczenie
   4. Specyficzność
3. **Bootstrap**
   1. kiedy i dlaczego używać
4. **SASS**
   1. po co używać
   2. loader do reacta
   3. co ma ponad css
   4. tworzenie plików css preprocesorem
   5. zobaczyć LESS – co to i podstawowe różnice do SASS
   6. SCSS
5. **GIT i jego funkcje**
   1. Mergowanie
   2. Branchowanie
   3. pull requesty
   4. cofanie się do commitów
   5. Stash
   6. Konflikty - skąd się biorą i jak je rozwiązywać
   7. poprosić Ozza
   8. git flow
6. **Npm, a yarn**
   1. popytać Ozza
7. **JS**
   1. Ify
   2. Pętle
   3. Funkcje
   4. Prototypy
   5. operatory logiczne
   6. undefined vs null
   7. Asynchroniczność
8. **JQuery**
   1. Funkcjonalności
   2. dlaczego wypadł z łask
   3. zalety na swój czas
9. **OOP**
   1. dlaczego obiektowość
   2. implementacja w react
10. **Ajax**
    1. ajax a redux
    2. po co używać
    3. zady i walety
11. **API**
    1. api na stronach
    2. api do mongo w mern
    3. hash użytkownika API
12. **Node**
    1. przypomnieć sobie jak postawić serwer na czystym node
    2. różnice z php
    3. back-end – serwer i zapytania do niego
13. **ReactJS**
    1. szablony (PuG EJS)
    2. struktura danych
    3. cykl życia komponentu
    4. ReactDOM
    5. state vs props
    6. dlaczego React
    7. podstawowe różnice z AgularJS
14. **ES6**
    1. co wprowadza
    2. jaki sposób kompiluje się do es2015 (babel)
15. **Webpack**
    1. Ozz
    2. Pluginy
    3. Loadery
    4. HMR
16. **ExpressJS**
    1. Middleware
    2. routing – endpointy
    3. CRUD
    4. JSON w JS
    5. autoryzacja
    6. serwowanie plików
17. **Inne**
    1. Makiety
    2. SCRUM
    3. Polling
    4. Streaming
    5. Long-polling
    6. Socket.io
18. **Redux**
    1. Flux
    2. Cqrs
    3. event sourcing
    4. zasady korzystania
    5. programowanie funkcyjne
    6. store w mern
    7. React-redux
    8. middleware redux
    9. data flow
19. **React-router**
    1. co to jest i jak działa
20. **MongoDB**
    1. nierelacyjne bazy danych
    2. SQL
    3. Mongoose
    4. operacje na bazie Mongo
21. **MERN**
    1. inne wersje np. MEAN
    2. kompletnie inne technologie PERL
    3. aplikacje izomorficzne
    4. server side rendering
    5. wirtualizacja (VBox, vagrant, docker) – Ozz
    6. generator kodu
    7. Redux-thunk
    8. Fetch
    9. przejrzeć dokumentację
22. **Pozostałe**
    1. REST API
    2. normalizacja danych
    3. ReactDnD
    4. compose i connect
    5. obadać punkty połączenia frontu z back-endem
23. **Zasady**
    1. DRY
    2. KISS
    3. poziomy abstrakcji
    4. SOLID Principles
24. **Wzorce projektowe**
25. **Migracje**
26. **Zagadnienia serwerowe**
    1. deployowanie, CI/CD
    2. Cache’owanie (memcache, redis, cloudflare)
    3. Domeny (DNS, Vhosty)