

e-Škole

USPOSTAVA SUSTAVA RAZVOJA
DIGITALNO ZRELIH ŠKOLA
(PILOT PROJEKT)

CARNET
znanje povezuje

Tehnička dokumentacija o izvornim datotekama i
elementima e-Škole digitalnog obrazovnog
sadržaja (DOS) za nastavne predmete

Geografija za 5. razred osnovne škole i 1. razred
opće gimnazije

Povijest za 5. razred osnovne škole i 1. razred
opće gimnazije

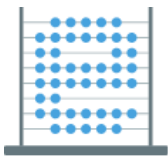
Fizika za 3. i 4. razred opće gimnazije

Kemija za 3. i 4. razred opće gimnazije



Sadržaj

1. UVOD	2
2. STRUKTURA PROGRAMSKOG KODA I ELEMENATA DIGITALNIH OBRAZOVNIH SADRŽAJA (ARHIVA IZVORNOG KODA)	3
2.1. OSNOVNE TEHNOLOGIJE NA KOJIMA POČIVA CIJELI DOS	3
2.2. STRUKTURA DIREKTORIJA	4
2.3. <i>BUILD</i> PROCES	5
<i>Webpack</i>	5
<i>Struktura JS datoteka</i>	6
<i>Gulp</i>	8
<i>Struktura SCSS datoteka</i>	8
<i>Ikone</i>	9
2.4. DODATNE SPECIFIČNOSTI.....	10
<i>Geogebra</i>	10
<i>KaTeX</i>	10
3. STRUKTURA DIREKTORIJA SADRŽAJNOG DIJELA (HPUB ARHIVA)	11
4. DIZAJN I VIZUALNI STANDARD	15
4.1. BOJE	15
<i>Fizika</i>	15
<i>Geografija</i>	15
<i>Kemija</i>	15
<i>Povijest</i>	15
4.2. TIPOGRAFIJA.....	15
4.3. IKONE I PIKTOGRAMI	16
4.4. PRISTUPAČNOST.....	16



1. Uvod

U ovom dokumentu iznijet će se sve informacije koje su potrebne da bi se programski kôd i datoteke DOS-a mogli koristiti, razumjeti, održavati i nadograđivati. Za to će biti bitno razumjeti strukturu programskog koda, ali i sadržajnog dijela te dijelove gdje se to dvoje preklapa. Objasnit će se tehnologije koje su korištene kao i razlozi za njihov odabir.



2. Struktura programskog koda i elemenata digitalnih obrazovnih sadržaja (arhiva izvornog koda)

2.1. Osnovne tehnologije na kojima počiva cijeli DOS

- HTML - <https://www.w3.org/html/>
- CSS - <https://www.w3.org/Style/CSS/Overview.en.html>
- JavaScript - <https://www.w3schools.com/js/DEFAULT.asp>
- SVG - <https://www.w3.org/TR/SVG11/>
- VueJS (v2.6.10) - <https://vuejs.org/>
- SCSS (v7.1.0) - <https://sass-lang.com/>
- Gulp (v3.9.1) - <https://gulpjs.com/>
- Webpack - <https://webpack.js.org/>
- NodeJS (v12.19.0) - <https://nodejs.org/en/>
- NPM (v6.14.8) - <https://www.npmjs.com/>

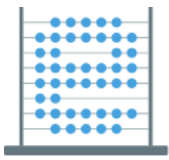
Datoteka package.json sadrži još precizniji uvid u *libraryje*, *frameworke* i pakete koji se koriste:

```
"devDependencies": {
  "@babel/cli": "^7.10.1",
  "@babel/core": "^7.10.2",
  "@babel/preset-env": "^7.4.3",
  "babel-loader": "^8.0.5",
  "css-loader": "^2.1.1",
  "eslint": "^6.8.0",
  "eslint-plugin-vue": "^6.2.1",
  "extract-text-webpack-plugin": "^3.0.2",
  "fancy-log": "^1.3.3",
  "fs": "0.0.1-security",
  "gulp": "^3.9.1",
  "gulp-autoprefixer": "^6.0.0",
  "gulp-babel": "^8.0.0",
  "gulp-clean-css": "^3.10.0",
  "gulp-color": "0.0.2",
  "gulp-concat": "^2.6.1",
  "gulp-if": "^2.0.2",
  "gulp-include": "^2.3.1",
  "gulp-merge": "^0.1.1",
  "gulp-newer": "^1.4.0",
  "gulp-plumber": "^1.2.1",
  "gulp-postcss": "^8.0.0",
  "gulp-rename": "^1.4.0",
  "gulp-sass": "^3.2.1",
```



```
"gulp-sourcemaps": "^2.6.5",
"gulp-strip-css-comments": "^1.2.0",
"gulp-stylus": "^2.7.0",
"gulp-svg-sprite": "^1.5.0",
"gulp-svg-sprites": "^4.1.1",
"gulp-tap": "^1.0.1",
"gulp-uglify": "^2.1.2",
"gulp-uncss": "^1.0.6",
"gulp-util": "^3.0.8",
"mkdirp": "^0.5.1",
"postcss-reporter": "^6.0.0",
"postcss-scss": "^2.0.0",
"sass-loader": "^7.1.0",
"vue-loader": "^14.2.2",
"vue-slick": "^1.1.15",
"vue-style-loader": "^4.1.2",
"vue-template-compiler": "^2.6.10",
"webpack": "^4.29.6",
"webpack-cli": "^3.3.0",
"webpack-dev-server": "^3.3.1",
"webpack-merge-and-include-globally": "^2.1.16",
"yargs": "^12.0.5"
},
"dependencies": {
  "@babel/polyfill": "^7.4.3",
  "axios": "^0.20.0",
  "core-js": "^3.0.1",
  "dateformat": "^3.0.3",
  "focus-visible": "^5.1.0",
  "gulp-require-tasks": "^1.2.1",
  "intersection-observer": "^0.7.0",
  "jquery": "^3.4.0",
  "lodash-es": "^4.17.15",
  "lozad": "^1.9.0",
  "natives": "^1.1.6",
  "screenfull": "^4.2.0",
  "vue": "^2.6.10",
  "vue-axios": "^2.1.5"
}
```

2.2. Struktura direktorija



Sve datoteke navedene u ovom dijelu sačinjavaju elemente koji su nužni za funkcioniranje DOS-ova, ali ne uključuju nikakve sadržajne datoteke uz iznimku GeoGebra elemenata, a svi ostali elementi sadržaja se nalaze u arhivama DOS-ova.

/src – u ovom direktoriju nalaze se sve izvorne datoteke programskog koda. Dva glavna poddirektorija su **carnet** i **common**, oni su komplementarni i ovisni jedan o drugome.

/src/common – sadrži općenite dijelove koda, SCSS i JS datoteke koje se koriste u DOS temi.

/src/carnet – sadrži specifične dijelove koda za funkcioniranje DOS teme, konfiguraciju *build* alata, kao i sve *asete*, a to uključuje:

- **/src/carnet/config** – konfiguracija build alata (Webpack)
- **/src/carnet/fonts** – svi fontovi koji se koriste
- **/src/carnet/icons** – izvorne SVG datoteke koje se *buildaju* u ikone koje se koriste u DOS-ovima
- **/src/carnet/images** – grafičke datoteke koje se koriste u DOS-ovima, ovdje je riječ o elementima sučelja i slično, ne o sadržajnim datotekama
- **/src/carnet/scripts** – sve JS datoteke
- **/src/carnet/scss** – sve SCSS datoteke
- **/src/carnet/twig** – dijelovi Twig *templatea* koji se koriste za kreiranje konačnog HTML-a DOS-ova.

/dist – u ovom direktoriju nalaze se sve kompajlirane datoteke koje se u konačnici koriste pri prikazu DOS-ova kao i neki *3rd party* dodaci korišteni za prikaz sadržaja.

/dist/GeoGebra – sadrži *3rd party* kod koji služi za prikaz interaktivnih elemenata razvijenih kroz GeoGebra platformu.

/dist/geogebra-src – izvorne GGB datoteke koje se mogu uređivati kroz GeoGebra aplikaciju.

/dist/katex – sadrži *3rd party* kod koji služi za prikaz LaTeX formula.

node_modules – sadrži sve pakete iz **package.json** kao i sve njihove međuovisnosti, međutim postaje vidljiv tek nakon inicijalizacije *build* procesa koji je opisan kasnije u tekstu.

package.json – ova datoteka sadrži popis paketa koji su nužni za rad s izvornim kodom.

README.md – ova datoteka sadrži kratke upute za programera koji će raditi na kodu.

2.3. Build proces

Za početak rada potrebno je pokrenuti naredbu `npm install`. Ona će instalirati sve potrebne pakete za funkcioniranje *build* procesa, te u tom trenutku **node_modules** direktorij postaje dostupan.

Webpack

Glavni dio build procesa kontrolira Webpack i to kompajliranje JS koda, *single file* Vue komponenti te konkatenciju raznih JS modula u 4 konačna *bundlea*:

- **/dist/bundle.js** – sadrži kod VueJS komponenti
- **/dist/common.js** – sadrži sav kod koji nije u vidu VueJS komponenti



- /dist/vendor.js – sadrži sav *3rd party* kod
- /dist/vue.js – sadrži VueJS framework.

Konfiguracija webpacka se nalazi u **/src/carnet/config/webpack.config.carnet.js** (razvojna verzija) i u **/src/carnet/config/webpack.config.carnet.production.js** (produksijska verzija). Razlika između razvojne i produkcijske konfiguracije Webpacka je što produkcijska kreira datoteke s .min sufiksom u imenu, a JS sadržaj je minificiran. Naredbom `npm run carnet` pokreće se Webpack u razvojnoj verziji, *watcher* prati promjene nad datotekama i na svaku promjenu pokreće kompajliranje koda.

Naredba `npm run carnet_prod` pokreće buildanje koda za produkcijsku verziju, ovaj proces može potrajati desetak sekundi.

Struktura JS datoteka

Generalno JS kod je pisan koristeći najmoderniju ES6 sintaksu i mogućnosti, a koristeći Babel *plugin* za Webpack se taj kod konvertira u stariju inaču EcmaScripta koju stariji preglednici mogu interpretirati.

Glavna odlika arhitekture koda jeste hibridna VueJS aplikacija, koja se definira u jednoj datoteci, a koristi primarni *inline template* definiran u HTML-u koji je onda kroz VueJS obogaćen funkcionalnostima koristeći direktive i prilagođene komponente. Dodatno uvelike se oslanja na uključivanje raznih dodatnih VueJS komponenti, koje su većinom pisane u SFC (single file component) formatu sa .vue ekstenzijom, i poneku komponentu koja se opet oslanja na *inline template* i .js ekstenziju.

Sav VueJS kod u konačnici završi u `bundle.js` datoteci, a za njeno izvršavanje je preduvjet učitavanja `vue.js`, `common.js` i `vendor.js` datoteka. Iz tog razloga se sve skripte poslije `vue.js` u HTML-u uključuju sa `defer` atributom.

JS i CSS kod pojedinih interaktivnih komponenti se na stranicu uključuje *inline* i to u procesu *exporta*, odnosno izrade `hpub` arhive. Takve komponente su izvedene na način da svoju konfiguraciju spremaju na globalnu varijablu, te se ta varijabla koristi za registraciju i inicijalizaciju globalne VueJS komponente, koja se onda konzumira u *inline templateima* u nastavku HTML-a.

Ovakvom arhitekturom osigurana je relativno malena veličina *bundlea* koji su učitani na svim stranicama s obzirom na funkcionalnosti, a kod za pojedine bogate interaktivne komponente se učitava po potrebi samo na stranicama na kojima se te komponente koriste.

/src/carnet/scripts/main.js – glavna JS datoteka, njen zadatak je inicijalizacija vršne VueJS komponente kao i pozivanje koda koji uključuju Pojmovnik, Tražilicu, renderiranje LaTeX jednadžbi, interaktivnost korisničkog sučelja DOS-a van interaktivnih elemenata, što uključuje razne galerije, *carousele*, *lightboxove*, *scrollanje* na klik, *custom* video player rješenje i slično.



Prvi dio main.js datoteke sadrži razne *polyfills* koji omogućavaju korištenje modernih JS mogućnosti u starijim preglednicima, konkretno IE.

U nastavku je vidljivo pozivanje raznih VueJS mixina koji se koriste u vršnoj komponenti:

initMediaHub – kreira instancu VueJS komponente koja je zadužena za to da na stranici DOS-a samo jedan video/audio player može biti aktivan u svakom trenutku. Funkcionira na način da pri početnom učitavanju stranice na svaki audio i video element dodaje *event listener* koji brinu o tome da ako se jedan element pokrene, pauzira sve ostale.

initLatex – definira konfiguraciju za Katex (<https://katex.org/>) koji služi za prikaz LaTeX formula u DOS-ovima. Koristi se autorender plugin <https://katex.org/docs/autorender.html>, definiraju *delimiteri* unutar kojih će se nalaziti LaTeX izrazi, te posebne klase unutar kojih će se ignorirati latex izrazi. Naime Katex zna ulaziti u konflikte sa VueJS template *renderingom* ako se ne konfigurira da ignorira neke dijelove DOM-a.

initSearch – dodaje u DOM skriptu za pretraživanje sadržaja DOS-ova. Budući da se indeks po kojem se vrši pretraživanje generira tek pri *exportu*, odnosno izradi arhive, nužno je da se tražilica uvjetovano uključuje na stranicu.

scrollToBlock – metoda koju vršna komponenta koristi da bi korisniku pomaknula scroll poziciju preglednika do točno određenog bloka, ovisno o zapisu u URL-u.

initSideControls – koristi IntersectionObserver kako bi na blokovima koji se nalaze u *viewportu* prikazao kontrole za inkluzivni prikaz bloka i alternativni audio sadržaj.

U nastavku datoteke definirane su globalne VueJS komponente koje se koriste.

video-player – budući da se HTML datoteke DOS-ova mogu pokretati u raznim okruženjima, uključujući direktno s diska, a bez poslužitelja, web preglednici nameću određena ograničenja na korisnika primarno radi sigurnosti. Jedno od tih ograničenja jeste blokiranje XHR mrežnih upita. Budući da za učitavanje titlova u HTML video se koristi XHR, korišteno je rješenje koje se oslanja na to da je sadržaj titlova ispisan u sami HTML. Pri učitavanju videa, SRT titlova se učitava iz DOM-a i koristeći WebVTTParser se kreira traka s titlovima koja se onda učitava na video element.

block-image – komponenta zadužena za rudimentarni prikaz slika na stranicama, kao i za napredniji prikaz gdje se slika može otvarati u *fullscreen* prikazu, imati naslov, opis i dodatne metapodatke. Konfiguracija se radi u HTML-u koristeći VueJS props.

izzi-icon – komponenta za prikaz SVG ikone. Koristi VueJS props kako bi se definiralo koja ikona i u kojoj dimenziji se želi prikazati.



izzi-block – bazna generička komponenta koja služi kao wrapper za svaki blok koji se vidi na stranici, od najjednostavnijih, do najkompleksnijih. Ova komponenta brine o prikazu blokova, inkluzivnom sadržaju blokova, alternativnom audio sadržaju, omogućava da se korisniku namjesti scroll tako da vidi točno određeni blok i slično.

glossary – komponenta koja brine o prikazu Pojmovnika, povezivanju pojedinih pojmova sa slovom, *scrollanjem* do pozicije za određeno slovo i sve ostalo vezano za Pojmovnik.

analytics – *renderless* komponenta koja na Google Analytics šalje podatke o korištenju DOS-ova, prati pokretanje/zaustavljanje video *playera*, pregledavanje jedinica, otvaranje/zatvaranje galerija slika i pokušaje rješavanja kvizova.

Gulp

Dodatno uz korištenje Webpacka za JS, koristi se i Gulp, primarno za kompajliranje SCSS datoteka te za izradu ikona od SVG-ova. Pokretanjem naredbe gulp u direktoriju **/src/carnet** pokreće se *watcher* koji prati promjene nad SCSS datotekama i na svaku promjenu pokreće kompajliranje koda. Za Gulp ne postoje odvojene razvojne i produkcijske verzije, već se producirani CSS automatski minificira, a *source-maps* su dostupne kako bi se olakšao razvoj.

SCSS datoteke su logički grupirane na više razina i ima ih jako velik broj, no objasniti će se osnovni principi i najvažnije pojedine datoteke. Svaka SCSS datoteka treba imati prefix `_` prije nego li se uveze u **style.scss** kako bi se ispravno kompajlirala kao *partial*, a ne samostalna CSS datoteka. SCSS je pisan imajući na umu BEM metodologiju (<https://en.bem.info/methodology/>), nužno je napomenuti da ima i komada *3rd party* koda koji je pisan drugačijom metodologijom. U takvim slučajevima pokušao se poštovati stil autora koji je taj kôd izvorno pisao, a opet ostati dosljedan vlastitoj metodologiji koja je ipak varijacija na BEM, gdje se koristi `__` kao separator u imenima blokova i elemenata, a `--` kao *modifier* za varijacije boje, veličine i tomu sličnog.

Struktura SCSS datoteka

Kao osnovica za stilove korišten je Bootstrap, odnosno pojedine njegove komponente, kako bi se ubrzao razvoj i postigao ujednačeni prikaz u različitim web preglednicima.

```
// Mixins
@import "../common/scss/mixins/general-mixins";
@import "../common/scss/mixins/breakpoints";

// Utilities
@import "../common/scss/utilities/functions";
@import "../common/scss/utilities/variables";
@import "../common/scss/utilities/reboot";
@import "../common/scss/utilities/utilities";
```



```
@import "../common/scss/utilities/flex";
@import "../common/scss/utilities/spacing";
@import "../common/scss/utilities/embed";
@import "../common/scss/utilities/debug";

// Layout mixins
@import "../common/scss/mixins/grid-framework";
@import "../common/scss/mixins/grid";
```

Dodatne bitnije SCSS datoteke su:

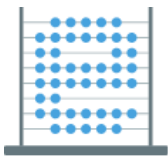
- **/src/carnet/scss/style.scss** – glavna SCSS datoteka koja poziva sve ostale.
- **/src/carnet/scss/_general.scss** – općeniti stilovi.
- **/src/carnet/scss/theme/_palettes.scss** – varijacije boja za razne DOS-ove.
- **/src/carnet/scss/theme/_variables.scss** – razne varijable pomoću kojih se može značajno prilagoditi izgled DOS-ova.
- **/src/carnet/scss/type/_font-face.scss** – @font-face definicije web fontova koje se koriste u DOS-ovima.
- **/src/carnet/scss/settings/_dark-mode.scss** – razne prilagodbe za prikaz u noćnom prikazu.
- **/src/carnet/scss/settings/_dyslexic.scss** – razne prilagodbe za prikaz uz font za disleksiju.
- **/src/carnet/scss/components/*** – razni SCSS-ovi za prikaz komponenti sučelja.
- **/src/carnet/scss/templates/*** – razni SCSS-ovi za prikaz pojedinih *templatea* DOS-ova.

Ikone

Generiranje ikona koristeći gulp ne događa se automatizmom kao koajliranje SCSS-ova, već se pokreće naredbom gulp svgicons, kada se pokaže potreba za doradom seta ikona. Preduvjet za funkcioniranje ove naredbe je da se ikone kao individualne SVG datoteke postave u direktorij **/src/carnet/icons**, a direktorij **/src/carnet/icons/_build** isprazni. Nakon pokretanja gulp svgicons, u **/src/carnet/icons/_build/symbol/svg** direktoriju možemo pronaći SVG datoteku čiji sadržaj treba kopirati u **/src/carnet/twig/functions/icons.twig**.

Također **/src/carnet/icons/_build/symbol/ sprite.symbol.html** sadrži primjer korištenja ikona kao i CSS potreban za njihovo funkcioniranje. Upravo taj generirani inline CSS treba iskopirati u datoteku **/src/carnet/scss/icons/_icons.scss** kako bi se ikone prikazivale u točnoj dimenziji i kako bi naslijedivale boje postavljene u CSS-u.

Kod pripreme ikona bitno je znati da dimenzija treba biti 24x24px, a ne bi trebalo koristiti stroke ni fill attribute, naime njihova vrijednost se postavlja CSS-om, a ne *inline*. Ovakav pristup omogućava da postoji samo jedna od svake pojedine ikone, i CSS-om se regulira prikaz, a ne da se mora proizvesti svaka varijacija posebno. Naziv SVG



datoteke u izvornom direktoriju će biti naziv po kojemu će se ikona kasnije pozivati u HTML-u.

2.4. Dodatne specifičnosti

Geogebra

Za neke od interaktivnih elemenata u DOS-ovima korištena je aplikacija GeoGebra. Izvorne datoteke se nalaze u dostavljenoj arhivi izvornog koda. Bitno je napomenuti da se dotične datoteke u HTML uključuju kao base64 enkodirani stringovi *inline*, budući da zbog načina distribucije DOS-ova postoji mogućnost da se sadržaji pokreću bez poslužitelja, a što znači da preglednik blokira XHR upite. *Inline* uključivanjem GGB datoteka zaobilazi se ovo ograničenje, međutim ukoliko se želi editirati izvorna GeoGebra potrebno je base64 enkodiranu datoteku vratiti u izvorni oblik, koristeći npr. alat <https://www.base64decode.org/>. Ovaj postupak će isproducirati ZIP paket, koji se može otpakirati, datoteke uređivati, a nakon uređivanja jhe potrebno napraviti obrnuti postupak i sadržaj paketa ZIP-ati, vratiti natrag u base64 format koristeći npr. alat <https://www.base64encode.org/> i potom isproducirani kod zamijeniti u HTML-u.

Alternativno, i dapače poželjno je uređivati paket koristeći službeni GeoGebra alat na <https://www.geogebra.org/>. U tom slučaju ZIP paket koji se dobije nakon base64 dekodiranja treba preimenovati na način da dobije ekstenziju .ggb. Takav paket može se otvoriti u GeoGebra alatu, urediti po želji, *exportati*, ponovo base64 enkodirati i ubaciti u HTML stranicu.

KaTeX

Rješenje odabrano za prikaz LaTeX formula jeste <https://katex.org/>, odabrano je spram "klasičnog" MathJax rješenja zbog drastično povećanih performansi učitavanja, što se naročito osjeti na mobilnim i tablet uređajima. Vokabular podržanih LaTeX izraza je vidljiv na poveznici <https://katex.org/docs/supported.html>. Više detalja o konkretnoj implementaciji je rečeno pri opisu **initLatex** mixina, najvažnije je imati na umu da ako se koristi u VueJS komponentama, treba paziti da se dijelove koje će VueJS u *templateima* popularirati s nekakvim vrijednostima modela izolira od autorender Katex plugina, na način da im se postavi klasa **js-katex-ignore**.



3. Struktura direktorija sadržajnog dijela (HPub arhiva)

Kako bi struktura direktorija i datoteka bila razumljiva i programskim parserima, a u svrhu distribucije sadržaja koristi se hpub standard <https://github.com/bakerframework/baker/wiki/hpub-specification>.

To podrazumijeva postojanje **book.json** datoteke u root direktoriju u kojoj su navedeni relativne poveznice do svih sadržaja. Evo i primjera:

```
{
  "title": "Fizika 3",
  "author": "CARNET",
  "url": "none",
  "date": "30-09-2020 05:09:10",
  "contents": [{
    "url": ".V10972.html",
    "title": "Fizika 3"
  }, {
    "url": ".V10973.html",
    "title": "Harmonijsko titranje"
  }, {
    "url": ".V10974.html",
    "title": "Harmonijsko titranje i jednoliko gibanje po kru\u017enici"
  }, {
    "url": ".V10975.html",
    "title": "Titranje tijela na opruzi"
  }, {
    "url": ".V10976.html",
    "title": "Matemati\u010dko njihalo"
  }, {
    "url": ".V13485.html",
    "title": "Izmjerimo ubrzanje sile te\u017ee pametnim telefonom"
  }, {
    "url": ".V10977.html",
    "title": "Energija titranja"
  }, {
    "url": ".V10978.html",
    "title": "Prisilno i prigu\u0161eno titranje"
  }, {
    "url": ".V12136.html",
    "title": "Aktivnosti za samostalno u\u010denje"
  }, {
    "url": ".V10979.html",
    "title": "Test za procjenu usvojenosti ishoda"
  }
]
```



```
}, {  
  "url": ".V10980.html",  
  "title": "Valovi"  
}, {  
  "url": ".V10983.html",  
  "title": "Nastanak i opis mehani\u010dkog vala"  
}, {  
  "url": ".V10984.html",  
  "title": "Brzina \u0161irenja vala"  
}, {  
  "url": ".V10985.html",  
  "title": "Odbijanja i lom valova"  
}, {  
  "url": ".V10986.html",  
  "title": "Interferencija i ogib valova"  
}, {  
  "url": ".V12137.html",  
  "title": "Aktivnosti za samostalno u\u010denje"  
}, {  
  "url": ".V13477.html",  
  "title": "Projekt model vala"  
}, {  
  "url": ".V12485.html",  
  "title": "Seizmograf"  
}, {  
  "url": ".V10987.html",  
  "title": "Test za procjenu usvojenosti ishoda"  
}, {  
  "url": ".V10981.html",  
  "title": "Zvuk"  
}, {  
  "url": ".V10988.html",  
  "title": "Nastanak zvu\u010dnog vala"  
}, {  
  "url": ".V10989.html",  
  "title": "Razina zvuka"  
}, {  
  "url": ".V10990.html",  
  "title": "Stojni val"  
}, {  
  "url": ".V10991.html",  
  "title": "Dopplerov u\u010dinak i zvu\u010dno one\u010di\u0161enje"  
}, {  
  "url": ".V18217.html",
```



```
"title": "Projekt 2. Zvu\u010dno one\u010di\u0161\u0107enje u mojoj \u0161koli"  
, {  
  "url": ".V18219.html",  
  "title": "Projekt 3. Zvu\u010dno one\u010di\u0161\u0107enje u mojem gradu i okolici"  
, {  
  "url": ".V12140.html",  
  "title": "Aktivnosti za samostalno u\u010denje"  
, {  
  "url": ".V10992.html",  
  "title": "Test za procjenu usvojenosti ishoda"  
, {  
  "url": ".V10982.html",  
  "title": "Geometrijska optika"  
, {  
  "url": ".V10993.html",  
  "title": "Zakoni geometrijske optike"  
, {  
  "url": ".V10994.html",  
  "title": "Zrcala"  
, {  
  "url": ".V10995.html",  
  "title": "Le\u0107e"  
, {  
  "url": ".V10996.html",  
  "title": "Razlaganje svjetlosti"  
, {  
  "url": ".V10997.html",  
  "title": "Opti\u010dki instrumenti"  
, {  
  "url": ".V12141.html",  
  "title": "Aktivnosti za samostalno u\u010denje"  
, {  
  "url": ".V10998.html",  
  "title": "Test za procjenu usvojenosti ishoda"  
, {  
  "url": ".V13493.html",  
  "title": "Footer"  
, {  
  "url": ".V13494.html",  
  "title": "Impresum"  
, {  
  "url": ".V13715.html",  
  "title": "Sidebar"  
, {
```



```
"url": ".V29294.html",  
"title": "Upute za uporabu"  
}, {  
"url": ".V13716.html",  
"title": "Pojmovnik"  
}  
}
```

U *root* direktoriju se nalazi datoteka **index.html** i **Ovdje pokrenite.html** koje su sadržajno istovjetne, a obje se mogu koristiti za pokretanje DOS-a.

/datastore – sadrži sve multimedijalne datoteke koje se koriste u DOS-ovima. Datoteke su podijeljene u direktorije prvo po jedinstvenom identifikatoru sadržaja u kojem se nalaze, zatim po tipu sadržaja, i zatim po datumu, npr. **/datastore/18/publication/10964/pictures/2020/01/22**.

/profil/dist – sadrži sve kompajlirane JS i CSS datoteke, kao i sve potrebne grafičke *asete*, fontove, itd. Ukoliko dolazi do kakvih funkcionalnih dorada isporučenih arhiva sadržaja, upravo ovaj direktorij bi se trebao zamijeniti s **dist** direktorijem iz arhive koda.

Ostatak sadržaja direktorija je popisan u datoteci **book.json**.



4. Dizajn i vizualni standard

4.1. Boje

Dizajn DOS-ova izveden je u 4 primarne palete boja, kako bi se postigla distinkcija između DOS-ova različitih predmeta. Svaka paleta definirana je s dvije primarne boje koje su međusobno komplementarne i koriste se za naglašavanje pojedinih dijelova sučelja poput *call-to-actiona*, klikabilnih elemenata, *hero* sekcija itd.

Fizika

Primarna boja: #f43d98

Sekundarna boja: #0f80c1

Geografija

Primarna boja: #f4b13d

Sekundarna boja: #61b021

Kemija

Primarna boja: #3de9f4

Sekundarna boja: #b04921

Povijest

Primarna boja: #ffbd00

Sekundarna boja: #0fc1c1

Ostatak boja u sučelju dijeljen je između svih DOS-ova, a najviše je riječ o neutralnim bojama kao što su razne nijanse sive, bijela, crna, i boje za univerzalno označavanje točnosti rješenja zadatka i to zelena #a5da6f za točno i crvena #cd001c za netočno.

4.2. Tipografija

Kod odabira tipografije vodilo se računa o sljedećim aspektima:

- mora podržavati veliki broj slovnih znakova i tipografskih elemenata
- mora podržavati znakove iz različitih jezika i pisama
- mora biti jednostavna i čitka
- mora biti lako i uvijek dostupna
- obitelj mora sadržavati što veći broj rezova zbog različitih primjena.

Odabrana tipografija koja zadovoljava navedeno je IBM Plex Sans. Dostupna je besplatno na <https://fonts.google.com/specimen/IBM+Plex+Sans>.

“Fall back” tipografija koja se koristi ako postoji problem s učitavanjem fontova s interneta je bilo koja sans-serifna tipografija koja je postavljena na korisnikovom računalu kao osnovna (npr. Arial, Verdana, Helvetica...).



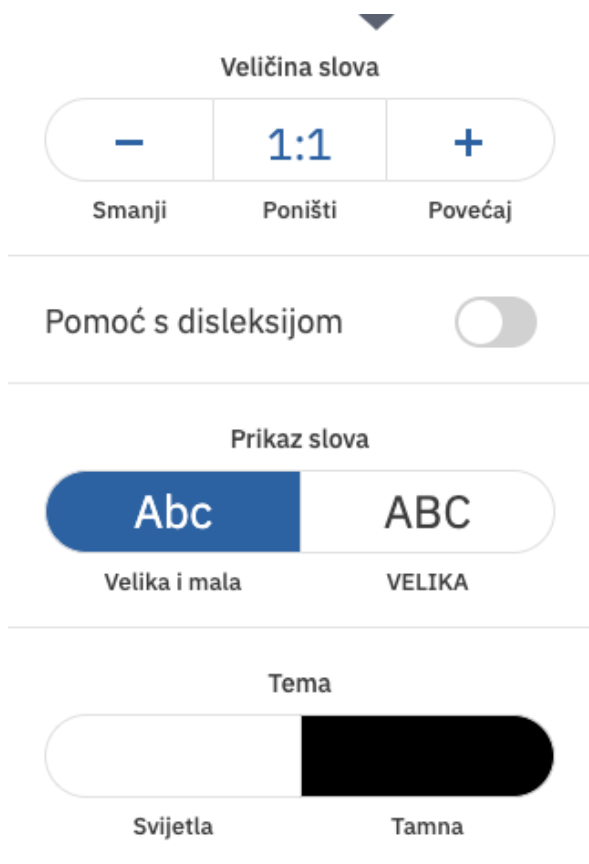
4.3. Ikone i piktogrami

U sučelju dodatno značenje i/ili uočljivost nekim elementima dana je koristeći ikone uz tekstualne oznake. Ikone koje se koriste su primarno inspirirane Material Icons setom od Googlea, primarno zbog svoje univerzalne prepoznatljivosti, neutralnog estetskog stila i otvorenog izvornog koda, što je omogućilo da ih se koristi kao SVG-ove kroz ranije opisani *build* proces radi postizanja boljih performansi učitavanja i bolje pristupačnosti.

Dostupne su besplatno na <https://material.io/resources/icons/?style=baseline>.

4.4. Pristupačnost

Pristupačnost je aspekt dizajna prožet svugdje, a uključuje brigu o dovoljno jakim kontrastima boja, čitkoj tipografiji, razmaku između elemenata i položaju na stranici, vidljivom označavanju *hover* i *focus* stanja kako bi korisniku bilo jasno kakvu interakciju može očekivati na stranici prije nego li se ona dogodi. Dodatno uz to korisniku se dozvoljava da sam po želi podešava pojedine aspekte sučelja.



Korisnik sam može:

- povećati sve tekstualne elemente (to uključuje sve veličine naslova, podnaslova, tekstualnih blokova, potpisa pod slikama, videima, galerijama itd.),
- zamijeniti tipografiju za onu prilagođenu za osobe s disleksijom,
- omogućiti prikaz isključivo u velikim slovima radi lakše čitljivosti,
- uključiti noćni prikaz u kojemu je povećan kontrast između elemenata, a dodatno se smanjuje opterećenje na oči u večernjim satima kada je osvjetljenje slabije.