



Europska unija  
Zajedno do  
fondova EU



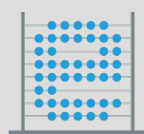
# Tehnička dokumentacija

UGOVOR O NABAVI USLUGA IZRADE OTVORENIH DIGITALNIH OBRAZOVNIH SADRŽAJA

GRUPA 8. Izrada otvorenih digitalnih obrazovnih sadržaja za prvi i drugi razred opće gimnazije za predmet biologiju te priručnika za njihovu primjenu za nastavnike, br. 258-100-830/17



Europska unija  
Zajedno do fondova  
EU



**CARNET**  
znanje povezuje

Projekt je sufinancirala  
Europska unija iz Europskog  
socijalnog fonda.

Više informacija o EU fondovima  
možete pronaći na:  
[www.strukturnifondovi.hr](http://www.strukturnifondovi.hr)



## Sadržaj

<b>1. Uvod</b>	<b>4</b>
<b>2. Struktura programskog koda i elemenata digitalnih obrazovnih sadržaja</b>	<b>5</b>
A. Osnovne tehnologije na kojima počiva cijeli DOS	5
B. Struktura direktorija	6
C. HPub struktura	8
<b>3. Struktura direktorija sadržajnog dijela</b>	<b>11</b>
A. Build proces - Inicijalizacija	11
B. Build proces - Gulp	11
C. SCSS	11
I. Struktura SCSS direktorija i datoteka	12
D. Ikone	13
E. JavaScript	14
F. Build proces - Browserify	14
II. Generalni naputci pri razvoju novih komponenti	16
III. Razni primjeri komponenti	16
IV. Popis komponenti po jedinicama	18
<b>4. Dizajn i vizualni standard</b>	<b>22</b>
A. Boje	22
B. Tipografija	24
C. Ikone i piktogrami	25
I. Digitalni obrazovni sadržaji	25
II. Moduli	26
III. GUI (grafičko korisničko sučelje)	27
IV. Pristupačnost	28
D. Mreža / Grid	29
E. Multimedijски elementi	30
I. Videozapisi	30
II. Fotografije	30
<b>5. Mobilne aplikacije</b>	<b>31</b>

A. Korištena tehnologija	31
B. Razvojna okolina	32
C. Nadogradnja i izmjene aplikacija	36
I. Tehnički preduvjeti	36
D. Popis aplikacija	37
<b>6. Priručnici</b>	<b>38</b>

# 1. Uvod

U ovom dokumentu iznijet ćemo sve informacije koje su potrebne da bi se programski kôd i datoteke DOS-a dostavljene u sklopu nabave 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. Objasniti ćemo tehnologije koje se koriste kao i razloge za njihov odabir.

Uz ovaj dokument dostavljamo i sljedeće priloge:

- Prilog 1 - Arhive DOS-a

U ovoj mapi se nalaze arhive cjelovitih DOS-eva Biologija 1 i Biologija 2 te pojedinačnih modula

- Prilog 2 - Priručnici

U ovoj mapi se nalaze:

- Izvorna datoteka cjelovitog priručnika
- Pdf datoteke priručnika (cjeloviti priručnik i priručnici uz pojedine module)
- Epub datoteke priručnika (cjeloviti priručnik i priručnici uz pojedine module)
- Word datoteka cjelovitog priručnika

- Prilog 3 - Tehnička dokumentacija

U ovoj mapi se nalazi:

- Dokument tehničke dokumentacije
- Arhiva dizajna, odnosno izvorne datoteke dizajn elemenata

- Prilog 4 - Arhive multimedijских i interaktivnih elemenata

U ovoj se mapi nalaze:

- fotografije
- video
  - videozapisi
  - titlovi
- interaktivni elementi

## 2. Struktura programskog koda i elemenata digitalnih obrazovnih sadržaja

### A. Osnovne tehnologije na kojima počiva cijeli DOS

- HTML - <https://www.w3.org/html/>
- CSS - <https://www.w3.org/Style/CSS/Overview.en.html>
- JS - <https://www.javascript.com/>
- SVG - <https://www.w3.org/TR/SVG11/>
- vue.js (v. 2.3.4.) - <https://vuejs.org/>
- SCSS - <https://sass-lang.com/>
- Gulp - <https://gulpjs.com/>
- NodeJS (v. 8.9.4) - <https://nodejs.org/en/>
- NPM (v. 5.6.0) - <http://npmjs.com/>
- Browserify - <http://browserify.org/>

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

```
"gulp": "^3.9.1",  
"gulp-autoprefixer": "^2.3.1",  
"gulp-concat": "^2.6.1",  
"gulp-csso": "^1.1.0",  
"gulp-header": "^1.8.8",  
"gulp-jade": "^1.1.0",  
"gulp-notify": "^2.2.0",  
"gulp-plumber": "^1.1.0",  
"gulp-rename": "^1.2.2",  
"gulp-sass": "^3.0.0",  
"gulp-svg-sprite": "^1.3.6",  
"gulp-uglify": "^1.5.4",  
"gulp-util": "^3.0.7",  
"marked": "^0.3.6",  
"orchestrator": "^0.3.8",  
"replace-ext": "0.0.1",  
"tiny-lr": "^0.1.7",  
"vinyl": "^0.5.3",
```

```
"babel-core": "^6.0.0",
"babel-preset-es2015": "^6.0.0",
"babelify": "^7.2.0",
"browserify": "^13.0.1",
"browserify-hmr": "^0.3.1",
"cross-env": "^1.0.6",
"envify": "^3.4.1",
"http-server": "^0.9.0",
"npm-run-all": "^2.1.2",
"uglify-js": "^2.5.0",
"vueify": "^9.1.0",
"watchify": "^3.4.0"
"babel-polyfill": "^6.26.0",
"chart.js": "^2.6.0",
"dragsterjs": "^1.5.0",
"gulp-sourcemaps": "^2.6.0",
"interactjs": "^1.3.3",
"jquery": "^3.2.1",
"lodash": "^4.17.4",
"nouislider": "^10.0.0",
"simplebar": "^2.5.1",
"vinyl-buffer": "^1.0.0",
"vinyl-source-stream": "^1.1.0",
"vue": "^2.3.4",
"vue-chartjs": "^2.7.2",
"vue-perfect-scrollbar": "0.0.14",
"vue-slick": "^1.1.6",
"vue2-selectize": "^1.0.3"
```

## B. Struktura direktorija

/ - Na početnoj razini se nalazi index.html, on služi isključivo za pretpregled elemenata tijekom razvoja. Ovdje također nalazimo **README.md** - sažet dokument koji služi da bi se *developer* koji se prvi put susreće s kodom mogao započeti s radom bez nedoumica.

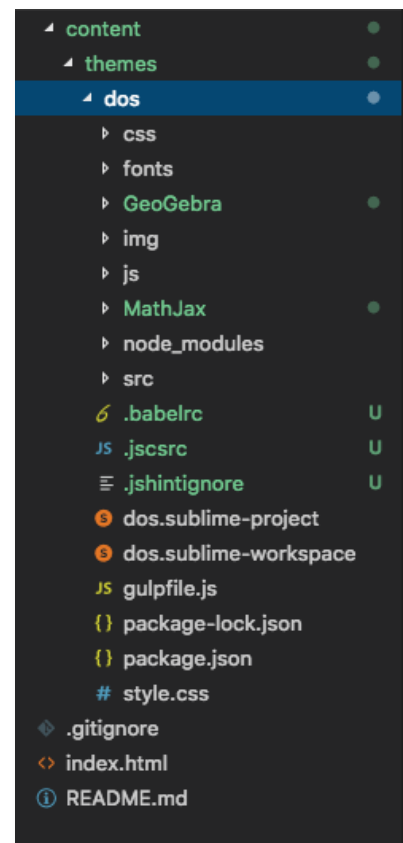
**/content/themes/dos** - U ovom direktoriju se nalazi sav programski kod, izvorne datoteke ali i konačni kompajlirani rezultati izvornih datoteka. Jako bitna datoteka je **package.json**, u kojoj su navedeni paketi koji su nužni da bi cijeli *build* proces radio kao i dio potrebne konfiguracije. **node\_modules** direktorij sadrži upravo pakete koji su navedeni u **package.json**, kao i njihove međuovisnosti, međutim on će biti vidljiv tek nakon inicijalizacije *build* procesa kojeg ćemo

naknadno detaljnije opisati, zajedno sa **gulpfile.js** koja je glavna konfiguraciona datoteka *build* alata Gulp.

**/content/themes/dos/src** - Ovdje nalazimo izvorne datoteke koje su prekursori za sav JS i CSS kôd kao i izvorne SVG datoteke koji se koriste kao ikonice u raznim interaktivnim elementima ali i elementima sučelja DOS-a. Detaljnije ćemo pojasniti sadržaj svakog poddirektorija u opisu *build* procesa.

**/content/themes/dos/src/img** - Ovdje su sadržane sve slikovne datoteke koje se koriste u elementima sučelja i interaktivnim elementima. Poddirektoriji su strukturirani na način da prate što je bliže moguće strukturu **/content/themes/dos/src/js/**

**components** - kako bi pronalazak datoteka koje su vezane za pojedine interaktivne elemente bio što lakši. U radu smo pokušali što više slikovnih datoteka koje se tiču elemenata sučelja izvesti koristeći SVG kako bi postigli što je moguće bolji prikaz na uređajima s raznim veličinama i rezolucijama zaslona, a kako bi veličina datoteka ostala što manja. U slučajevima gdje to nije bilo moguće izvesti, korišteni su PNG ili JPG formati. U svim slučajevima datoteke su prije korištenja maksimalno optimizirane kako bi se smanjila veličina bez vidnog gubitka kvalitete. Ovaj korak je odrađen u dizajn dijelu procesa razvoja, a koristeći softver ImageOptim (<https://imageoptim.com/mac>) te SVGO GUI (<https://github.com/svg/svgo-gui>) za vektorske datoteke.



**/content/themes/dos/MathJax** sadrži *3rd party* kôd koji služi za prikaz formula u LATEXu. **/content/themes/dos/Mathjax/config/default.js** je konfiguraciona datoteka koju koristimo, u njoj je bitna stavka registracija mchem ekstenzije za prikaz kemijskih jednadžbi, na liniji 108 `"[Contrib]/mhchem/mhchem.js"`,

Dio konfiguracije MathJaxa se nalazi inline u HTML datotekama, a odgovoran je za prikaz formula u istom fontu kao i ostatak sadržaja, ovdje ga navodimo u cijelosti.

```
<script type="text/x-mathjax-config">
  MathJax.Hub.Config({
    "HTML-CSS": {availableFonts: ["TeX"]},
    MMLorHTML: {prefer: "HTML"}
  });
  MathJax.Hub.Register.StartupHook("HTML-CSS Jax Ready",function () {
    var VARIANT = MathJax.OutputJax["HTML-CSS"].FONTDATA.VARIANT;
    VARIANT["normal"].fonts.unshift("MathJax_SansSerif");
    VARIANT["bold"].fonts.unshift("MathJax_SansSerif-bold");
    VARIANT["italic"].fonts.unshift("MathJax_SansSerif-italic");
    VARIANT["-tex-mathit"].fonts.unshift("MathJax_SansSerif-italic");
    document.body.classList.add('mathjax-rendered')
  });
  MathJax.Hub.Register.StartupHook("SVG Jax Ready",function () {
    var VARIANT = MathJax.OutputJax.SVG.FONTDATA.VARIANT;
```

```

    VARIANT["normal"].fonts.unshift("MathJax_SansSerif");
    VARIANT["bold"].fonts.unshift("MathJax_SansSerif-bold");
    VARIANT["italic"].fonts.unshift("MathJax_SansSerif-italic");
    VARIANT["-tex-mathit"].fonts.unshift("MathJax_SansSerif-italic");
  });
</script>

```

**/content/themes/dos/GeoGebra/** sadrži *3rd party* kôd koji služi za prikaz interaktivnih elemenata razvijenih kroz *GeoGebra* platformu. U **/content/themes/dos/GeoGebra/src** se nalaze izvorne GGB datoteke *GeoGebra* objekata koji se koriste u interaktivnim elementima. Njih je moguće kroz *GeoGebra* aplikaciju uređivati i mjenjati.

**/content/themes/dos/fonts** - sadrži fontove koji se koriste u cijelom sučelju DOS-a i interaktivnim elementima. Primarno *Roboto* sa svim svojim varijacijama, a tu je i *Open Dyslexic* koji postaje dostupan u DOS-u odabirom opcije iz izbornika za pristupačnost.

## C. HPub struktura

Kako bi struktura direktorija i datoteka bila razumljiva i programskim parserima, a u svrhu distribucije sadržaja koristimo 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": "Biologija 1",
  "author": "Željko Krstanac, Karlo Horvatin",
  "url": "",
  "contents": [
    "index.html",
    "./biologija-1/m01/",
    "./biologija-1/m01/kazalo-multimedije/",
    "./biologija-1/m01/j05/",
    "./biologija-1/m01/j04/",
    "./biologija-1/m01/j03/",
    "./biologija-1/m01/j02/",
    "./biologija-1/m01/j01/",
    "./biologija-1/m01/pojmovnik/",
    "./biologija-1/m02/",
    "./biologija-1/m02/kazalo-multimedije/",
    "./biologija-1/m02/pojmovnik/",
    "./biologija-1/m02/j09/",
    "./biologija-1/m02/j08/",
    "./biologija-1/m02/j07/",
    "./biologija-1/m02/j06/",
    "./biologija-1/m02/j05/"
  ]
}

```



```

    "/biologija-1/m02/j04/",
    "/biologija-1/m02/j03/",
    "/biologija-1/m02/j02/",
    "/biologija-1/m02/j01/",
    "/biologija-1/m03/",
    "/biologija-1/m03/kazalo-multimedije/",
    "/biologija-1/m03/pojmovnik/",
    "/biologija-1/m03/j07/",
    "/biologija-1/m03/j06/",
    "/biologija-1/m03/j05/",
    "/biologija-1/m03/j04/",
    "/biologija-1/m03/j03/",
    "/biologija-1/m03/j02/",
    "/biologija-1/m03/j01/",
    "/biologija-1/m04/",
    "/biologija-1/m04/kazalo-multimedije/",
    "/biologija-1/m04/pojmovnik/",
    "/biologija-1/m04/j05/",
    "/biologija-1/m04/j04/",
    "/biologija-1/m04/j03/",
    "/biologija-1/m04/j02/",
    "/biologija-1/m04/j01/",
    "/biologija-1/m05/",
    "/biologija-1/m05/kazalo-multimedije/",
    "/biologija-1/m05/pojmovnik/",
    "/biologija-1/m05/j03/",
    "/biologija-1/m05/j02/",
    "/biologija-1/m05/j01/",
    "/biologija-1/impresum/",
    "/biologija-1/m05/"
]
}

{
  "title": "Biologija 2",
  "author": "Željko Krstanac, Karlo Horvatin",
  "url": "",
  "contents": [
    "index.html",
    "/biologija-2/m01/",
    "/biologija-2/m01/kazalo-multimedije/",
    "/biologija-2/m01/j05/",
    "/biologija-2/m01/j04/",
    "/biologija-2/m01/j03/",
    "/biologija-2/m01/j02/"
  ]
}

```

"/biologija-2/m01/j01/",  
"/biologija-2/m01/pojmovnik/",  
"/biologija-2/m02/",  
"/biologija-2/m02/kazalo-multimedije/",  
"/biologija-2/m02/pojmovnik/",  
"/biologija-2/m02/j03/",  
"/biologija-2/m02/j02/",  
"/biologija-2/m02/j01/",  
"/biologija-2/m03/",  
"/biologija-2/m03/kazalo-multimedije/",  
"/biologija-2/m03/pojmovnik/",  
"/biologija-2/m03/j03/",  
"/biologija-2/m03/j02/",  
"/biologija-2/m03/j01/",  
"/biologija-2/m04/",  
"/biologija-2/m04/kazalo-multimedije/",  
"/biologija-2/m04/pojmovnik/",  
"/biologija-2/m04/j03/",  
"/biologija-2/m04/j02/",  
"/biologija-2/m04/j01/",  
"/biologija-2/m05/",  
"/biologija-2/m05/kazalo-multimedije/",  
"/biologija-2/m05/pojmovnik/",  
"/biologija-2/m05/j07/",  
"/biologija-2/m05/j06/",  
"/biologija-2/m05/j05/",  
"/biologija-2/m05/j04/",  
"/biologija-2/m05/j03/",  
"/biologija-2/m05/j02/",  
"/biologija-2/m05/j01/",  
"/biologija-2/m06/",  
"/biologija-2/m06/kazalo-multimedije/",  
"/biologija-2/m06/pojmovnik/",  
"/biologija-2/m06/j16/",  
"/biologija-2/m06/j15/",  
"/biologija-2/m06/j14/",  
"/biologija-2/m06/j13/",  
"/biologija-2/m06/j12/",  
"/biologija-2/m06/j11/",  
"/biologija-2/m06/j10/",  
"/biologija-2/m06/j09/",  
"/biologija-2/m06/j08/",  
"/biologija-2/m06/j07/",  
"/biologija-2/m06/j06/",  
"/biologija-2/m06/j05/",

```
    "/biologija-2/m06/j04/",
    "/biologija-2/m06/j03/",
    "/biologija-2/m06/j02/",
    "/biologija-2/m06/j01/",
    "/biologija-2/impresum/",
    "/biologija-2/m06/"
  ]
}
```

### 3. Struktura direktorija sadržajnog dijela

Arhive sadržaja koje su dostavljene prate gore navedenu strukturu uz nekoliko razlika:

- ne sadrže **/content/themes/src** direktorij, već samo kompajlirane datoteke
- sadrže **/content/uploads** direktorij koji sadrži sve multimedijalne datoteke koje se koriste u sadržajima.
- sadrže direktorij s imenom DOS-a koji sadrži sve HTML sadržaje, a pokreće se klikom na **/index.html**

Za internu arhivu dostavljamo i kopije multimedijalnih sadržaja dostupnih u DOS-ovima. Podijeljeni su prvo prema nazivu DOS-a kojem pripadaju, a onda po kategoriji, multimedija/interaktivni elementi/.

#### A. *Build* proces - Inicijalizacija

Prva stvar koju je nužno učiniti da bi se počelo s razvojem je u direktoriju **/content/themes/dos/** pokrenuti naredbu **npm install**. To će instalirati sve potrebno za funkcioniranje build procesa, nakon čega će ranije spomenuti direktorij **node\_modules** postati dostupan.

#### B. *Build* proces - Gulp

#### C. SCSS

Gulp je zadužen primarno za kompajliranje SCSS datoteka u CSS. U tom procesu sve CSS datoteke iz **/content/themes/dos/src/scss** se kompajliraju u jednu datoteku **/content/themes/dos/style.css** koja se koristi u DOS-ovima. Ona sadrži sve stilove potrebne za prikaz DOS-ova, uz iznimku dijela stilova koji su definirani direktno u pojedinim interaktivnim elementima. U procesu kompajliranja također se automatizmom postavljaju potrebni vendor prefixi i rade optimizacije na CSS kodu kako bi prikaz među različitim inačicama preglednika bio zadovoljavajuć. Ovaj proces se odvija automatizmom nakon što se pokrene *watcher* naredbom

**gulp** u direktoriju **/content/themes/dos**, koji prati izmjene na svim SCSS datotekama i pri spremanju izmjena producira novi **style.css**.

## I. Struktura SCSS direktorija i datoteka

SCSS datoteke su logički grupirane na više razina i ima ih jako velik broj pa nećemo prolaziti svaku pojedinačno, no objasniti ćemo osnovne principe 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 smo pokušali poštovati stil autora koji je taj kôd izvorno pisao, ali opet ostati dosljedni vlastitoj metodologiji koja je ipak varijacija na BEM, gdje koristimo **\_** kao separator u imenima blokova i elemenata, a **--** kao *modifier* za varijacije boje, veličine i tomu sličnog.

- A. Kao osnovica za stilove, a i kostur *markupa* DOS-ova korišten je *HTML5 boilerplate* (<https://html5boilerplate.com/>) pa shodno tome datoteke **\_html5pb.scss** i **\_normalize.scss** uvozimo u **style.scss**. **\_normalize.scss** resetira *defaultne* stilove koje svaki preglednik ima na zajednički nazivnik, što nam omogućuje konzistentniji prikaz dizajna u raznim preglednicima i dugoročno olakšava pisanje SCSS-a. **\_html5pb.scss** u sličnoj maniri postavlja neke naše poželjne *defaulte* pa ih ne moramo iznova definirati kasnije u kodu.
- B. **\_main.scss** sadrži općenite stilove koji definiraju i oblikuju globalne elemente sučelja DOS-a
- C. **\_helpers.scss** sadrži razne *helper* klase koje u pravilu imaju jednu svrhu, a mogu se kombinirati da bi se dobili željeni efekti, pa se na taj način izbjegava pisanje dodatnih CSS selektora za elemente gdje je minimalna stilska intervencija potrebna.
- D. **\_variables.scss** sadrži razne SCSS varijable, što podrazumijeva definicije boja, dimenzija elemenata, media querija, fontova itd. Te varijable su globalno dostupne i koriste se u mnogim SCSS datotekama.
- E. **\_mixins.scss** sadrži razne SCSS mixine koji su pisani specifično za potrebe dizajna DOS-ova, kao i općenite mixine za npr. *clearanje floatova*, *image replacement* itd. Vjerovatno najčešće korišten je **brand()** koji omogućuje produciranje više varijanti DOS-ova u različitim bojama, njegovo je korištenje detaljnije opisano u samoj **\_mixins.scss** datoteci
- F. **\_fonts.scss** sadrži *font-face* definicije za Roboto i Open Dyslexic
- G. **\_icons.scss** sadrži općenite stilove za prikaz ikona koje Gulp generira
- H. **\_print.scss** stilovi za prikaz sadržaja DOS-ova u tisku
- I. **\_debug.scss** stilovi korišteni za *debugiranje* grešaka u *markupu*, elementi bez obaveznih atributa se bojuju žarko crvenom bojom da se lakše uoče
- J. **\_inclusive.scss** stilovi koji reflektiraju korisničke postavke koje se odabiru kroz izbornik za pristupačnost, veličina fontova, kontrastni i noćni mod
- K. **components/\*** - sadrži stilove za pojedine jednostavnije dijelove sučelja DOS-ova, obični *buttoni*, *inputi* i slični elementarni komadi sučelja su ovdje opisani
- L. **structures/\*** - sadrži stilove za kompleksnije dijelove sučelja, strukture su obično složene od više komponenti, zato radimo distinkciju

**M. interactive/\*** - pojedini stilovi nužni za funkcioniranje interaktivnih elemenata se ovdje nalaze jer ih nije bilo moguće postaviti direktno u **vue.js** komponentu

## D. Ikone

Sekundarna namjena Gulpa je produciranje ikona koje se koriste u elementima sučelja DOS-ova, a i u dobrom dijelu interaktivnih elemenata. Ovaj proces se ne dešava automatizmom kao kompajliranje SCSS-ova već se ručno pokreće kada dođe do potrebe da se set dostupnih ikona proširi novima ili modificira. Pokreće se naredbom **gulp sprites** u direktoriju **/content/themes/dos**. Nakon pokretanja gulp prolazi kroz sadržaj direktorija **/content/themes/dos/src/icons\_source**, taj direktorij naime sadrži isključivo SVG datoteke. Ime svake SVG datoteke odgovara imenu ikone koja će biti dostupna za korištenje u DOS-ovima. Rezultat pokretanja ove naredbe je vidljiv u direktoriju **/content/themes/dos/src/icons** gdje se nalazi jedna SVG datoteka (**sprite.symbol.svg**) koja sadrži sve ikone, kao i HTML datoteka (**sprite.symbol.html**) koja sadrži primjer korištenja ikona i potreban CSS za njihovo funkcioniranje. Nakon što ovaj proces završi sadržaj SVG-a valja kopirati u **<body>** HTML datoteka sa sadržajem DOS-a, idealno na sami početak kako bi se što prije učitale. Dodatno inline CSS koji se nalazi u **sprite.symbol.html** potrebno je iskopirati u **/content/themes/dos/src/\_icons.scss** kako bi se ti stilovi kompajlirali u konačni **style.css**.

Napomena: pri izradi SVG-ova za ikone, *fill* atribut ne smije biti postavljen, jer to osigurava da će ikonice boju naslijediti od svog parenta u DOM-u i da ih možemo koristiti u bilo kojoj boji bez da radimo više varijacija iste ikone samo radi promjene boje.

## E. JavaScript

U smislu JS-a, jedina funkcija Gulpa jest da kopira JS datoteke koje nije nužno kompajlirati iz /**content/themes/dos/src/js** direktorija u /**content/themes/dos/js**.

Ovdje su nam posebno zanimljive datoteke **main.js** i **plugins.js**, te direktorij **vendor**.

**main.js** ima puno zadataka, između ostaloga u njemu se nalazi kôd za sva ponašanja i interaktivnost korisničkog sučelja DOS-a van interaktivnih elemenata, što uključuje razne galerije, *carousele*, *lightboxove*, *scrollanje* na klik, *custom* video player rješenje i slično.

Možda najkompleksniji dio ove datoteke jest konfiguracija i inicijalizacija video *playera* koja počinje na liniji 320. Tu vidimo prvo unos stringova za lokalizaciju video *playera*, a nakon toga metodu koja se brine za to da bi titlovi radili. Titlovi se naime ispisuju u SRT formatu u DOM, iz DOM-a ih čitamo i parsiramo u VTT format, koristeći `WebVTTParser` koji kreira traku koju **video.js** plugin razumije i može prikazivati. Ovaj proces je jako bitan jer omogućuje učitavanje titlova bez korištenja AJAX zahtjeva koji su nam onemogućeni kada se DOS-u pristupa direktno preko HTML datoteka. Također, dio **main.js** koji se tiče video *playera*, a počinje na liniji 369 nakon inicijalizacije postavlja *event listener* na svaki *player* i brine se o tome da kad se jedan *player* pokrene, da se svi ostali pauziraju.

Ako se pojavi potreba za mjenjanjem titlova nekog videozapisa, u HTML-u treba zamjeniti SRT zapis.

**plugins.js** sadrži *3rd party* kôd kojeg se većinom iz **main.js** poziva i konfigurira. Uključuje rješenja za *carousel*, *lightbox* i video *player*.

**vendor** direktorij sadrži *libraryje* i *frameworke* koji su nam nužni, jQuery i Vue.js primarno, također i nekoliko plugina za *drag&drop* funkcionalnosti, za prikaz vremenske crte i slično. To su u pravilu paketi koji nisu bili dostupni kao moduli koje možemo uvesti direktno u komponente, pa se koriste na ovakav način.

## F. Build proces - Browserify

Drugi dio *build* procesa se odnosi na razvoj interaktivnih elemenata. Odabrana tehnologija za razvoje je Vue.js, ponajprije zbog podrške za *single-file* komponente (<https://vuejs.org/v2/guide/single-file-components.html>) koje omogućuju da se u jednoj datoteci sadrži sav kôd zadužen za funkcioniranje interaktivnog elementa, HTML i CSS za prikaz i JS za logiku i funkcioniranje. Dodatni razlog za korištenje **Vue.js** je jednostavnost korištenja, odnosno relativno blaga krivulja učenja korištenja, što znači da je developerima lakše uključiti se u rad na projektu i vrlo brzo početi producirati funkcionalne interaktivne elemente.

Da bi mogli koristiti *single-file* komponente nužno je imati *loader* koji razumije njihov sadržaj i transformira ga u čisti JS kôd kojeg preglednik može izvršavati. Za tu namjenu odabrali smo

*browserify*, sa dodacima *vueify* koji interpretira .vue datoteke, te *watchify* koji prati promjene na datotekama i kompajlira ih u **bundle.js**, koji na kraju sadrži sav JS kôd potreban za funkciniranje naših elemenata.

**/content/themes/dos/package.json** sadrži popis ovih paketa, kao i njihove konfiguracije i definicije skripti koje su nužne za razvoj. Važno je i napomenuti postojanje **/content/themes/dos/.babelrc** datoteke. Ona služi konfiguraciji babela (<https://babeljs.io/>) čija je namjena da ES6 kôd kakvim su pisani interaktivni elementi kompajlira u stariju verziju JS-a, a zadrži istu funkcionalnost. Ovo nam omogućava da developeri imaju više mogućnosti pri razvoju te da mogu pisati koncizniji, čišći kôd, a bez da gubimo išta na podršci za starije preglednike.

Za razvoj interaktivnih elemenata treba pokrenuti skriptu **npm run dev**. Ona pokreće kompajliranje Vue.js komponenti, a *output* se može naći u **/content/themes/dos/js/bundle.js**. Ta skripta također pokreće *live-reload* server, koji podržava *module injection*, što znači da će izmjene na elementima automatizmom nakon spremanja biti vidljive developeru u pregledniku, što značajno ubrzava razvoj, budući da kompajliranje cijelog **bundle.js** traje par sekundi, što je posljedica jako velikog broja interaktivnih elemenata koje imamo.

**npm run build** također kompajlira vue.js komponente, ali u **bundle.min.js**, taj proces obično traje duže od par sekundi budući da se pri kompajliranju radi i *uglify* odnosno komprimira kôd da bi se smanjila veličina fajla i ubrzalo učitavanje DOS-ova u konačnici.

Da bi *browserify* prepoznao komponentu i buildao je ona se mora registrirati. Taj dio se radi u **/content/themes/dos/src/js/app.js**, tako da se komponenta prvo importa, a zatim i registrira u dijelu gdje se inicijalizira vue.js aplikacija.

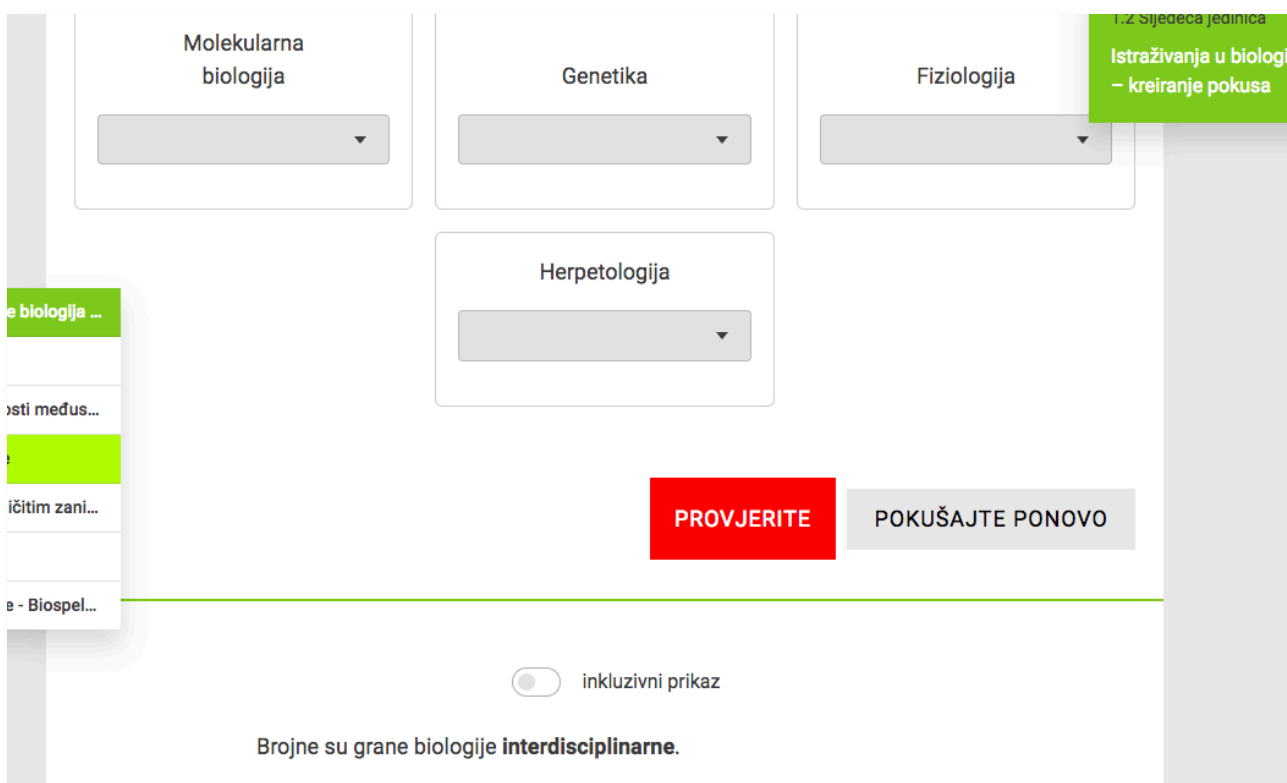
```
var app = new Vue({
  el: '#app',
  data: {
  },
  components: {
    'grada-i-uloga-vegetativnih-organa-biljke': GradaIUlogaVegetativnihOrganaBiljke
  },
  computed: {
  },
  methods: {
  }
});
```

PRIMJER REGISTRACIJE KOMPONENTE

## II. Generalni naputci pri razvoju novih komponenti

Pri razvoju novih komponenti treba se držati dosad postavljenih struktura i standarda. Da bi se zadržala vizualna konzistentnost treba koristiti `_variables.scss` i `_mixins.scss` koji se mogu importati u svaku komponentu. Korištenje tamo definiranih dimenzija garantira dobar prikaz na raznim veličinama ekrana, a boja garantira dovoljni kontrast u sučelju što je nužno za čitljivost materijala.

Treba također paziti da se zadovolje standardi pristupačnosti. S ovim može od pomoći biti ranije spomenut `_debug.scss` koji ako se omogući će žarko crvenom označiti buttone kojima fali type atribut i slike kojima fali alt atribut. Naravno uvijek na umu treba imati elementarna pravila pristupačnosti, koristiti pripadajuće elemente i attribute gdje god je to moguće. Ako se koriste *3rd party* rješenja u komponentama, paziti pri odabiru i da su ona pristupačna.



PRIMJER KAKO SE NEPRISTUPAČNI ELEMENTI OZNAČAVAJU KADA JE `_DEBUG.SCSS` AKTIVAN

## III. Razni primjeri komponenti

`/content/themes/dos/src/js/elements` sadrži sve interaktivne elemente, u direktoriju se nalaze globalno dostupni elementi, a u poddirektorijima s nazivima DOS-a i modula gdje se koriste se nalaze specifični elementi. Većina razvijenih elemenata jesu srednje razine interaktivnosti, a manji broj je visoke. Za srednju razinu interaktivnosti smo u radu prepoznali neke uzorke koji se ponavljaju te smo takve funkcionalnosti apstrahirali u `vue.js` komponente i `vue.js` mixine.



Odličan primjer takve komponente bi bio `/content/themes/dos/src/js/elements/drago.vue` i njen pripadajući mixin `/content/themes/dos/src/mixins/drago.js`. U samoj komponenti je definiran njen izgled i *template*, a u mixinu se nalazi sva potrebna logika za funkcioniranje. Ovaj način raspodjele kôda nam omogućuje da logiku iz ovog elementa koristimo i u drugima, a da ne ponavljamo isti kôd na više mjesta. Nisu svi elementi srednje razine interaktivnosti izvedeni na ovaj način, neki su ipak dovoljno drugačiji od svih ostalih da su iziskivali izradu prilagođenog prikaza i logike ali komentarima u samim komponentama su ostavljene korisne bilješke za developera koji odluči preuzeti ovaj kôd i raditi s njim dalje, pa se tako jednostavno može razumjeti odakle neka komponenta naslijeđuje prikaz ili funkcionalnost.

### **Komponente koje su dostupne za brzo kreiranje interaktivnih elemenata srednje razine interaktivnosti i njihovi kratki opisi:**

- `dd-drago.vue` - nudi rješavanje pitanja odabirom točnih odgovora iz dropdown izbornika
- `drago-columns` - drag&drop element u više stupaca
- `drago.vue` - osnovni drag&drop element
- `drago-group.vue` - drag&drop element s mogućnosti grupiranja odgovora
- `drago-slider.vue` - drag&drop element prikazan podijeljen u više koraka
- `drago-table.vue` - drag&drop element s popunjavanjem tablice
- `stepper.vue` - *wrapper* koji omogućuje podijelu kompleksnijih interaktivnih elemenata u više koraka
- `input-table.vue` - tablica sa pitanjima/odgovorima, nudi rješavanje unosom odgovora ili odabirom iz dropdown izbornika
- `input-boxes.vue` - mreža sa više pitanja s unosom odgovora ili odabirom iz dropdowna

### **Još neke zanimljive komponente i načini razvoja**

Dodatno bih istaknuli komponentu `/content/themes/dos/src/js/elements/int-header.vue` koja se koristi kao *child* komponenta svakog razvijenog elementa. Ona naime služi kao okvir unutar kojeg se komponente prikazuju, a također služi i kao *splash screen* za mobilne uređaje, gdje je prikaz interaktivnih elemenata odgođen dok se korisnik ne odluči za interakciju.

Komponenta `/content/themes/dos/src/js/elements/int-modal.vue` se koristi kao *child* komponenta mnogih drugih, a kao što se iz naziva da zaključiti, služi za prikaz modalnih dijaloga. U izvornom kôdu te komponente također se nalazi primjer korištenja, a vidljiv je i primjer korištenja *eventova* kako bi komponente međusobno komunicirale. U pravilu se radi na način *props down, events up*, *child* komponente dobijaju *properties* s kojima manipuliraju, a emitiraju *eventove* koje *parent* komponenta "sluša" i na njih reagira.

Komponenta `/content/themes/dos/src/js/elements/Flipcard.vue` je primjer komponente koja koristi *props*. Ovaj pristup nam omogućava da imamo beskonačno puno varijacija te komponente koje možemo generirati iz *backenda*. Još nekoliko primjera takvih komponenti:

- Absuto.vue - unos ispravnog odgovora
- Asoc.vue - rješavanje asocijacija
- Crosswordat.vue - križaljke
- Lisuto.vue - unos ispravnog odgovora
- Mflip.vue - otkrivanje sadržaja
- Note.vue - spremanje bilješki u *localStorage*
- Selector.vue - odabir točnih odgovora iz dropdown izbornika

**/content/themes/dos/src/js/components/quiz.js** je primjer komponente koja nije izvedena kao *single-file*, već je njen template *inline* u HTML datotekama.

Vue.js koristimo ne samo za interaktivne elemente, već i za neke dijelove sučelja DOS-ova, konkretno imamo:

- Accessibility.vue - služi za upravljanje izbornikom za pristupačnost.
- Breadcrumbs.vue - prikazuje breadcrumbs na jedinicama i modulima
- Scrollspy.vue i ScrollspyMM.vue - scrollspy element na jedinicama i Kazalu multimedije
- Search.vue - search input i dropdown u headeru
- Term.vue i Explanation.vue - pojmovi i pojašnjenja u sadržaju jedinica
- Overflow.vue - koristi se u sadržaju kad je širina sadržaja veća od širine ekraja uređaja. Tada ova komponente kreira *scrollbarove*.

#### IV. Popis komponenti po jedinicama

Svi interaktivni elementi u svrhu arhive dostavljeni su u izdvojenoj mapi naziva: Interaktivni elementi, dostavljeni s ovom tehničkom dokumentacijom.

Biologija 1		
Jedinica	Komponenta	Nasljeđuje/koristi
1.1	shema-prirodnih-znanosti	input-boxes
1.1	grane-biologije	input-boxes
1.1	bio-djelatnosti	drago-group
1.2	custo1	
1.2	timeline	
1.3	dijelovi-znanstvenog-rada	
1.3	brzina-reakcije	
1.4	zirafa-suncokret	modal.js
1.4	lisuto	

Biologija 1		
1.4	rod-vrsta-inicijal	modal.js
1.4	gljive-biljke-zivotinje	
1.4	umna-mapa-zivih-bica	
1.5	usloznjavanje-grade	drago
2.1	bez-vode-nema-zivota-ph	
2.2	ugljikohidrati-skriveni-u-namirnicama	modal.js
2.3	prehramdbena-piramida	drago-group
2.4	nukleinske-kiseline	stepper
2.4	adna	
2.5	djelovanje-enzima	
2.6	hranjive-tvari-u-namirnicama	input-boxes.js
2.7	mlijecna-staza	
2.8	na-granici-zivog-i-nezivog	drago
2.9	prepoznavanje-virusa	input-boxes
3.1	svjetlosni-mikroskop	dd-drago
3.1	mikroskopiranje	
3.2	grada-biomembrane	dd-drago
3.2	osmoza	
3.3	prokariotska-stanica	dd-drago
3.4	griffithov-pokus	
3.5	grada-eukariotske-stanice	dd-drago
3.5	usporedba-prokariotske-i-eukariotske-stanice	drago-slider
3.6	biljni-plastidi	input-boxes.js
3.7	razlicite-stanice-u-ljudskom-organizmu	input-boxes.js
4.1	povezanost-fotosinteze-i-stanicnog-disanja	drago
4.2	intenzitet-fotosinteze	
4.3	stanicni-ciklus-modal	stepper, modal.js
4.4	krosingover	
4.5	stanicni-ciklus	stepper, input-boxes.js
5.1	organogeneza	modal.js

Biologija 1		
5.2	vrste-ploda-kritosjemenjaca	drago-group
5.3	vrste-misicnog-tkiva	drago-group
5.4	organski-sustavi-zivotinja	

Biologija 2		
Jedinica	Komponenta	Nasljeđuje/koristi
1.1	interaktivna-karta	modal.js
1.1	dnd-kladogram	drago
1.1	kladogram-zivotinje	
1.2	systematics-quiz	
1.2	sistematika-lava	
1.2	hrast	drago-table
1.2	lijeska-organizacija	
1.3	zivotinje-dd	drago
1.3	absuto	
1.3	homo	drago-group
1.4	karta-hrvatske	
1.5	custo2	
2.1	virusi-dijelovi-bakteriofaga	drago.vue
2.2	patogene-bakterije	modal.js
2.3	bolesti-uzrokovane-virusima-i-bakterijama	input-table
3.1	morska-salata	drago
3.2	usporedba-protoktista	drago-slider
3.3	protoktisti	
4.1	kistac	modal.js
4.2	lisaj	modal.js
4.3	otrovne-gljive-hrvatske	modal.js
4.3	gljive	
5.1	tipovi-i-vrste-tkiva	drago-table

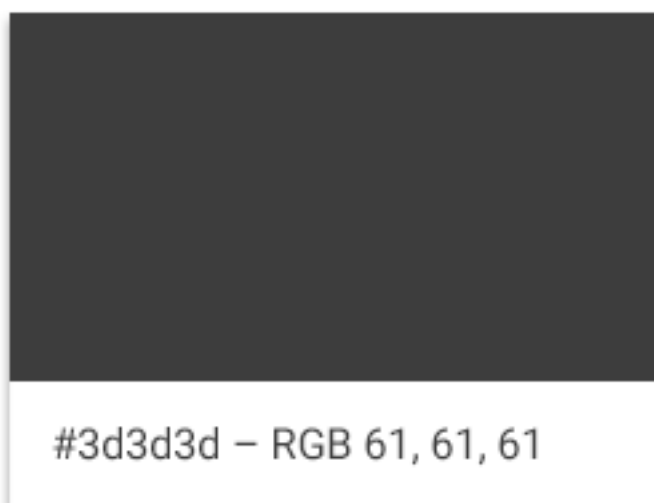
Biologija 2		
5.2	memo	
5.3	zivotni-ciklus-mahovina	modal.js
5.4	vrste-plodova	drago-table
5.5	usporedba-jednosupnica-i-dvosupnica	input-boxes.js
5.6	predstavniciflore	modal.js
5.7	usporedba-gametofita-i-sporofita	modal.js
6.1	simetrijazivotinje	modal.js
6.2	grada-tijela-spuzvi-i-koralja	drago-group.vue
6.3	zivotni-ciklus-govede-trakavice	modal.js
6.4	prehrana-puzeva-skoljkasa-i-glavonozaca	modal.js
6.5	zajednicke-prilagodbe-nametnickih-plosnjaka-i-oblica	modal.js
6.6	grada-gujavice	modal.js
6.7	preobrazbakukaca	drago
6.8	bodljikasi	dd-drago
6.9	usporedba-razlicitih-skupinaclankonozaca	drago-slider
6.10	svitkovci	dd-drago
6.11	usporedba-grade-riba-hrskavicnjaci-kostunjaca	modal.js
6.12	zivotni-ciklus-zabe	modal.js
6.13	poznatijezmije-hrvatske	input-boxes.js, modal.js
6.14	perje-ptica	modal.js
6.14	usporedba-ucinkovitostipluca-kraljeznjaka	
6.15	zubalo-biljojeda-i-mesojeda	modal.js
6.16	rodoslovno-stablo-svitkovaca	drago

## 4. Dizajn i vizualni standard

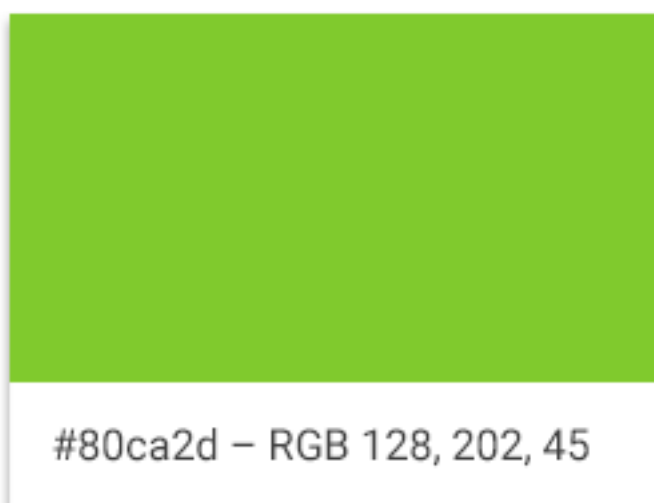
Slijedi kratko pojašnjenje dizajna i vizualnih standarda koji su korišteni u izradi DOS-a, svih sadržaja sučelja i sadržaja.

### A. Boje

Boja pozadine je tamno siva kako bi svi ostali detalji iskočili. Oslanjamo se na kolorističke i svjetlosne akcente i perspektive. Ono što je svjetlije ili u toplijoj boji čini se bliže. Na taj način imamo jasno izgrađenu strukturu i hijerarhiju objekata.



Svaki DOS ima definiranu glavnu boju koja određuje ton kompletnog proizvoda  
Za biologiju ta je boja zelena.

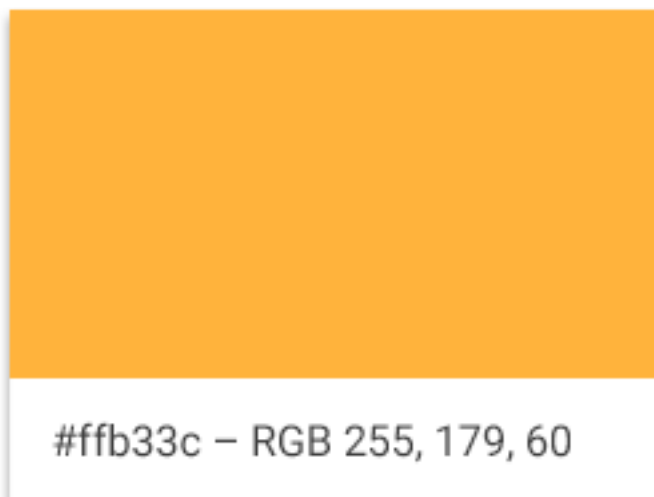


Uz glavnu boju tu je i dodatni ton koji služi za naglašavanje u detaljima.



Iz ovih boja izvedene su sve ostale nijanse koje se koriste u nekom postotku. Taj postotak ovisi o više varijabli: hijerarhijskom značenju i poziciji u odnosu na ostale elemente, boji pozadine i ostalih elemenata u okolini itd.

Osim ovih koriste se još dvije boje unutar sistema svih DOS-ova. One imaju specifične primjene i odabrane su iz toplog dijela spektra boja. Prva od njih je narančasta, koja se koristi na ikonama u stanju kada je pokazivač miša u poziciji preko ikone. Ona je odabrana jer je koloristički i svjetlosni akcent u odnosu na tamno sivu pozadinu, ali i na obje definirane boje DOS-ova (plavu i zelenu) koje su obje iz hladnog djela spektra boja.



No, zbog specifičnosti izrade ikona modula (layer styles, svjetlosni efekti, sjene, sjajevi, gradienti itd.) ovaj ton može se naći samo u bazi vektorskih objekata. Taj ton je izmijenjen stilovima layera u Photoshopu radi postizanja finalnog izgleda ikona.



Kod elemenata procjene znanja koristi se topla žuta nijansa. Isto tako, sve su procjene znanja u okvirima koji nemaju oštre vrhove. Tim sitnim detaljima umanjuje se nevoljkost ili strah učenika od pristupa tim elementima procjene znanja koji su u DOS-ovima osmišljeni kao dodatni edukativni elementi, a ne formalna provjera znanja.

## B. 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 sve ove uvjete je Roboto.

ABCČĆDĎEFGHIJKLMNOPQRSŠTUVWXYZŽabcčćd  
đefghijklmnopqrsštuvwxyzžАБВГГДЂЕЄЖЗСИІ  
ЙЈКЛЉМНЊОПРСТЋУЎФХЦЧЦШЩЪЫЬЭЮЯабв  
ггдђеєєжзсииійјклљмнњопрстћуўфхцчцшщъыь  
эюяАВГДЕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩαβγδεζηθικ  
λμνξοπρστυφχψωάΆέΈέΉήΊΐΌόΌύϋϋΥΎΩΆÂÊÔσϚă  
âêôσϚ1234567890'?''"!(%)[#]{@}/&\<-+÷x=>®©\$€  
£¥ç:;,.\*

Dostupna je besplatno na <https://fonts.google.com/specimen/Roboto>



“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...).

## C. Ikone i piktogrami

### I. Digitalni obrazovni sadržaji

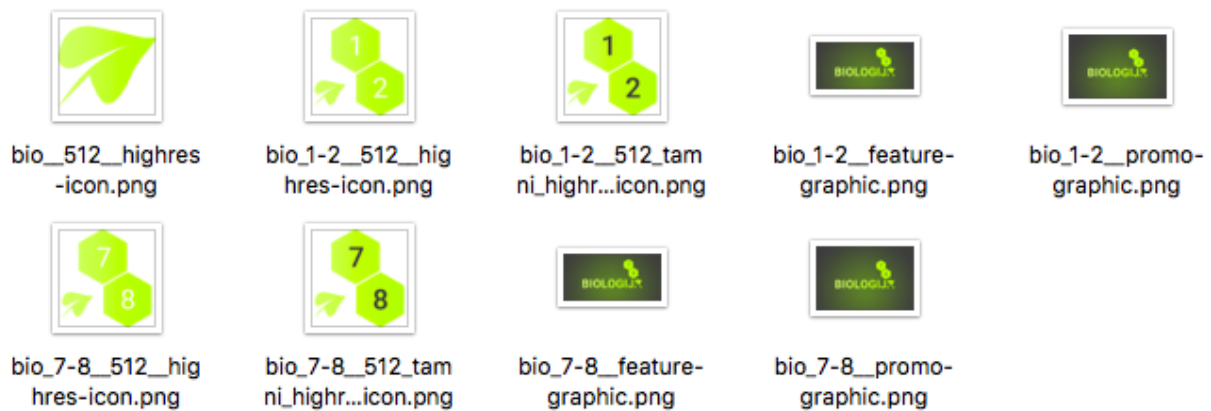
Distinkcija između DOS-ova izražena je u posebnim ikonama koje koristimo unutar samog naziva DOS-a.

Biologija je brandirana listom koji se nalazi unutar slova A. Trokutasta “rupa” unutar slova A je zamijenjena šesterokutom. Šesterokut u ovom slučaju predstavlja saće unutar košnice (škola kao košnica i učenici kao marljive pčelice) gdje sve ima svoj red, hijerarhiju, smisao i funkciju. U sklopu istog slova A nalazi se još jedan šesterokut unutar kojeg se nalazi broj koji korespondira s razredom kojemu je namijenjen ovaj DOS. 7 i 8 za sedmi i osmi razred osnovne škole, a 1 i 2 za prvi, odnosno drugi razred gimnazije.



Isti se objekt (list) koristi za ikone aplikacija i datoteka za preuzimanje.

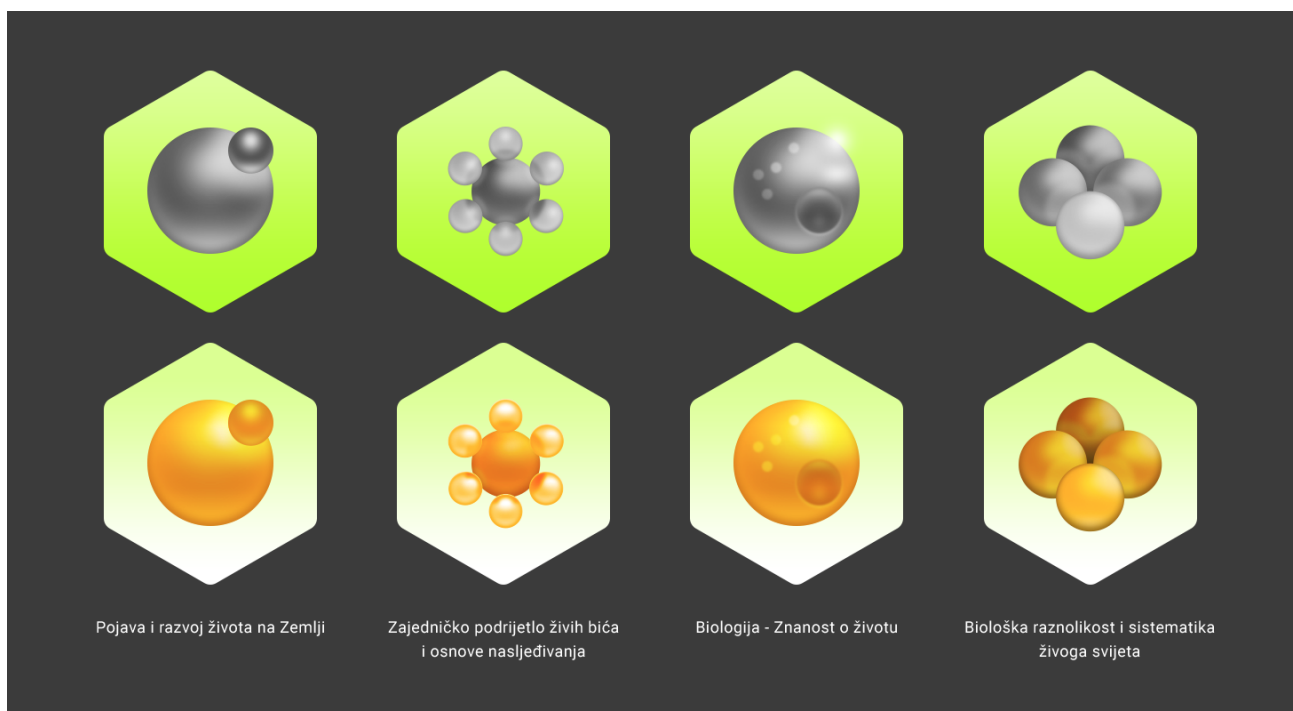
Ovo je samo jedan manji dio ikona i vizuala - ikone za aplikaciju na Android platformi. Ikone su napravljene u svim veličinama za primjenu na iOS, Android i Windows platformama.



## II. Moduli

Glavne ikone su svakako ikone modula.

Kako bi postojala jasna vizualna distinkcija između DOS-ova, iako su u identičnom okviru (šesterokutku) ikone se ipak razlikuju u obliku kojeg čini apstraktni prikaz sadržaja modula. ako je biologija prikazana metaforom atoma, kuglom. Unutar tih “kugli” nalaze se različiti drugi elementi - kružnice, elipse - ali svi ukupno čine to zadano tijelo kugle.



IKONE PRVIH MODULA IZ SVIH DOS-OVA BIOLOGIJE

U prvom redu su prikazane ikone u svojem normalnom stanju, a u drugom u svojem naglašenom stanju kada se pokazivačem miša prijeđe preko ikone.

### III. GUI (grafičko korisničku sučelje)

Ovo su ikone i piktogrami koje se koriste unutar grafičkog korisničkog sučelja.



PIKTOGRAM ZA PRISTUP OPCIJAMA PRISTUPAČNOSTI<sup>1</sup>

Odabran je prikaz ovog čovječuljka (inspiracija - Vitruvijev čovjek Leonarda da Vincija) zbog njegovog očitog prikaza čovjeka. Na njemu nema istaknutog prikaza posebnih potreba jer smatram da ih ne treba posebno isticati. Dapače, osobe s posebnim potrebama treba kao takve prihvatiti i olakšati im konzumiranje sadržaja. Upravo za to i služi sadržaj ovog izbornika.



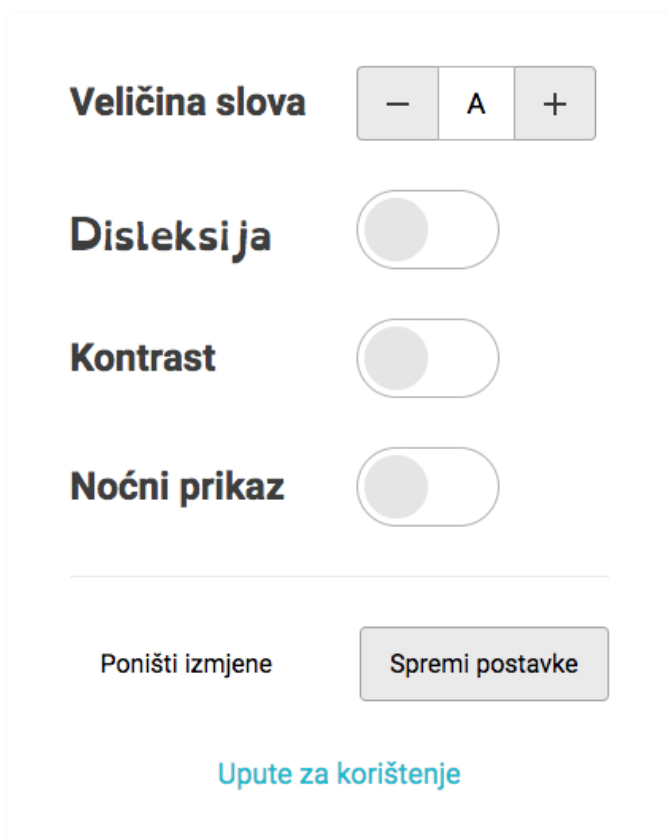
PIKTOGRAM ZA PRISTUP PROCJENI ZNANJA

Kod elemenata procjene znanja koristi se ova ikona. To je “stripovski” oblačić s upitnikom što simbolizira pitanje, a pošto se procjena znanja koristi na nivou modula, a ikone modula su definirane unutar oblika šesterokuta i ikone provjere znanja su također unutar šesterokuta kako bi se dodanto naglasila procjena sveobuhvatnog gradiva unutar dotičnog modula.

---

<sup>1</sup> Napomena - cijeli gumb je ovdje prikazan u duplo većoj veličini

## IV. Pristupačnost



The image shows a settings panel for accessibility. It contains four main options, each with a control element:

- Veličina slova**: A control with three buttons: a minus sign (-), the letter 'A', and a plus sign (+).
- Disleksija**: A toggle switch that is currently turned off.
- Kontrast**: A toggle switch that is currently turned off.
- Noćni prikaz**: A toggle switch that is currently turned off.

At the bottom of the panel, there are two buttons: "Poništi izmjene" (Reset changes) and "Spremi postavke" (Save settings). Below these buttons is a link labeled "Upute za korištenje" (Help).

Korisnik može povećati sve tekstualne elemente (to uključuje sve veličine naslova, podnaslova, tekstualnih blokova, potpisa pod slikama, videima, galerijama itd.), može zamijeniti tipografiju u onu prilagođenu za osobe sa disleksijom, može povećati kontrast ukupnog sadržaja (osim unutar slika i videa jer na to tehnički nemamo utjecaja) te uključiti noćni prikaz što je najkontrastnija varijanta koju smo predvidjeli - tekst u bijeloj boji prikazuje se na tamnoj pozadini što je znatno manje naporno za oči u večernjim satima kad je osvjetljenje slabije.

## D. Mreža / Grid

Popis jedinica jednog modula podijeljen je u tri stupca u širini od 914px što je i širina sadržaja svake jedinice. Naravno takva je situacija na većim ekranima. Pažnja je usmjerena u sredinu ekrana u širinu stupca optimalnu za čitanje.



**1.1**  
**Što je biologija i zašto je učimo?**

U uvodnoj jedinici DOS-a, u jednom školskom satu, baveći se predmetom proučavanja biologije i najvažnijim granama biologije, upoznat ćete se s elementima i načinima rada u DOS-u. Također, naglasak će biti na značaju dostignuća iz biologije za svakodnevni život i primjeni u

**1.2**  
**Istraživanja u biologiji – kreiranje pokusa**

Tijekom tri školska sata upoznat ćete osnovne korake znanstvene metode kroz formuliranje istraživačkih pitanja i hipoteza te kreirati nacrt istraživanja s kontrolnim i eksperimentalnim skupinama. Upoznat ćete i razliku između znanstvenih i stručnih

**1.3**  
**Istraživanja u biologiji – analiza rezultata**

Interpretirajući rezultate vlastitog i tuđih istraživanja, tijekom tri školska sata produbit ćete svoja znanja o znanstvenoj metodi u biologiji s naglaskom na analizu rezultata. Donijet ćete objektivne i logičke zaključke temeljene na rezultatima istraživanja. Također

Različite širine elemenata:

Moći ću	716 px
Tekst, fotografije, galerije, video	570 px
Interaktivni objekti	914 px
Flip card	750 px
Sadržaj unutar 3 stupca na modulu	824 px
Kviz	824 px
Sadržaj kviza	570 px

Sve izvorne datoteka korištene u digitalnim obrazovnim sadržajima, uključujući sučelje, aplikacije, sadržaj i ikone, nalaze se u mapi naziva *Dizajn - arhiva* dostavljenom s tehničkom dokumentacijom.

## **E. Multimedijски elementi**

Svi multimedijски elementi dostavljeni su u svrhu arhive u mapi Arhive multimedijских i interaktivnih elemenata.

Konkretno, multimedijски elementi se nalaze u mapama:

1. Video - sadrži videozapise, 2D i 3D animacije
2. Fotografije - sadrži fotografije, 2D i 3D ilustracije

### **I. Videozapisi**

Svaki videozapis unutar DOS-a formatiran je u video codecu h264, mp4 formatu, 1280x720 rezolucije sa stereo audiom (128 bit rate).

Podnapisi za videe su \*srt formata enkodirani kao UTF-8 i dostavljeni uz video arhive.

Podnapise je moguće isključiti i uključiti unutar same platforme, osim kod videa koji u svojim podnapisima imaju posebne znakove poput kemijskih formula koje sami titlovi ne podržavaju, pa su zbog točnosti zapisa ti titlovi montažom spojeni sa samim videom. Također, svaki video koji sadržava naraciju ima i pripadajući titl.

Audio svih videozapisa je montiran tako da se snimljena naracija za video u stereo formatu pušta samo s lijeve strane kako bi korisnici s raznim oštećenjima zvuka mogli lakše izolirati sadržaj ako im je to potrebno.

### **II. Fotografije**

U DOS-ovima su korištene fotografije iz sljedećih izvora:

1. GettyImages
2. Sciencephoto Library
3. Shutterstock
4. Pixabay
5. FreedImage

Za sve multimedijске sadržaje osigurana je ne-ekskluzivna, neprenosiva licenca za preuzimanje, čuvanje, reproduciranje, kopiranje, mijenjanje i prijenos sadržaja za neograničenu uporabu unutar konteksta digitalnih obrazovnih sadržaja.

## 6. Mobilne aplikacije

U sklopu nabave, dostavljen je i izvorni kôd za DOS mobilne aplikaciju za iOS, Android i UWP platforme. U sklopu kôda nalaze se 4 brancha koji su "brandirani" master branch, odnosno imaju grafičke elemente za pripadajuću grupu predmeta, tj. pripadajuću aplikaciju. Master branch sadržava sav temeljni kôd za funkcioniranje aplikacije, a pojedini branchevi na taj kôd dodaju brendiranje sukladno svakoj od četiri aplikacije. Na taj način od jednog kôda dobivamo vizualno i sadržajno četiri različite mobilne aplikacije.

Svaka mobilna aplikacija sadrži korisničko sučelje za pregled i preuzimanje cjelovitih DOS-eva te pripadajućih pojedinih modula.

Unutar aplikacije, DOS-evi i njihovi moduli mogu se pregledavati *online* ili se mogu preuzeti lokalno na uređaj kako bi se mogli koristiti *offline*, odnosno bez internetske veze.

Korisnik ima opciju preuzimanja/brisanja cjelovitog DOS-a ili preuzimanja/brisanja pojedinih modula. Ako korisnik klikne na opciju 'Preuzmi sve', preuzima se cjeloviti DOS sa linka na kojem je cjeloviti DOS, radi uštede memorijskog prostora. U tom slučaju korisnik ne može brisati pojedine module, nego briše sve, ako klikne na opciju 'Obriši sve.'

Prema navedenom scenariju, aplikacija preuzima ZIP datoteke s putanja dohvaćenih s CARNET-ovih servera te ih po preuzimanju raspakirava i omogućuje korisniku offline pregled materijala.

Svi elementi sučelja i interaktivnih elemenata u najvećoj mogućoj mjeri su razvijani responzivno, kako bi se mogli nesmetano koristiti i na manjim zaslonima ekrana. Osim smanjivanja pojedinih elemenata sučelja i sadržaja, ovisno o veličini ekrana mijenja se gdje god je to moguće i vrsta interakcije, na primjer element koji se na većim veličinama ekrana rješava metodom *drag&drop*, na mobitelu će se rješavati odabirom iz padajućeg izbornika, radi što boljeg korisničkog iskustva.

### A. Korištena tehnologija

Mobilne aplikacije izrađene su uporabom Xamarin.Forms *frameworka*. Xamarin.Forms je odabran jer omogućuje fleksibilnu i efikasnu izradu i kasnije održavanje nativnih mobilnih aplikacija u odnosu na razvoj aplikacije za svaku mobilnu platformu zasebno.

Za bazu podataka se koristi Akavache koji je baziran na SQLite. Odabran je zbog toga što je optimiziraniji i brži od konkurentskih rješenja te omogućuje 'in-memory cache'.

Za prikaz slika se koristi FFImageLoading jer omogućuje jednostavno upravljanje slikama i njihovo 'cacheiranje' (privremeno spremanje na disk).

Za preuzimanje velikih datoteka koristi se Xam.Plugins.DownloadManager jer podržava gore navedene zahtjeve za cjelovitim i modularnim preuzimanjem sadržaja.

Za modalne dijaloge se koristi standardni Acr.UserDialogs,.

Za *custom pop-up* notifikacije koristi se Rg.Plugins.Popup.

Programski jezik korišten za izradu aplikacija je C#.

## B. Razvojna okolina

- Visual Studio v15.7.5
- Xamarin.Android SDK v8.3.3.2
- Xamarin.Apple SDK v4.10.10.2
- Visual Studio for Mac v7.5.3
- Xcode v9.4.1

Android projekt:	
<package id="Acr.Support" version="2.1.0" targetFramework="monoandroid71" />	
<package id="Acr.UserDialogs" version="6.4.1" targetFramework="monoandroid71" />	
<package id="akavache" version="5.0.0" targetFramework="monoandroid80" />	
<package id="akavache.core" version="5.0.0" targetFramework="monoandroid80" />	
<package id="akavache.sqlite3" version="5.0.0" targetFramework="monoandroid80" />	
<package id="AndHUD" version="1.2.0" targetFramework="monoandroid71" />	
<package id="Microsoft.CSharp" version="4.4.1" targetFramework="monoandroid80" />	
<package id="Microsoft.NETCore.Platforms" version="2.0.1" targetFramework="monoandroid80" />	
<package id="Mindscape.Raygun4Net" version="5.5.4" targetFramework="monoandroid80" />	
<package id="NETStandard.Library" version="2.0.1" targetFramework="monoandroid71" />	
<package id="Newtonsoft.Json" version="10.0.3" targetFramework="monoandroid71" />	
<package id="Rg.Plugins.Popup" version="1.0.4" targetFramework="monoandroid71" />	
<package id="Rx-Core" version="2.2.5" targetFramework="monoandroid80" />	
<package id="Rx-Interfaces" version="2.2.5" targetFramework="monoandroid80" />	
<package id="Rx-Linq" version="2.2.5" targetFramework="monoandroid80" />	
<package id="Rx-Main" version="2.2.5" targetFramework="monoandroid80" />	
<package id="Rx-PlatformServices" version="2.2.5" targetFramework="monoandroid80" />	
<package id="Splat" version="1.6.2" targetFramework="monoandroid71" />	
<package id="SQLitePCLRaw.bundle_e_sqlite3" version="1.1.0" targetFramework="monoandroid80" />	
>	



**Android projekt:**

```
<package id="SQLitePCLRaw.core" version="1.1.0" targetFramework="monoandroid80" />
```

```
<package id="SQLitePCLRaw.lib.e_sqlite3.android" version="1.1.0" targetFramework="monoandroid80" />
```

```
<package id="SQLitePCLRaw.provider.e_sqlite3.android" version="1.1.0" targetFramework="monoandroid80" />
```

```
<package id="System.ComponentModel.TypeConverter" version="4.3.0" targetFramework="monoandroid71" />
```

```
<package id="System.IO.Compression" version="4.3.0" targetFramework="monoandroid71" />
```

```
<package id="System.IO.Compression.ZipFile" version="4.3.0" targetFramework="monoandroid71" />
```

```
<package id="System.Reflection.TypeExtensions" version="4.3.0" targetFramework="monoandroid71" />
```

```
<package id="System.Runtime.CompilerServices.Unsafe" version="4.4.0" targetFramework="monoandroid71" />
```

```
<package id="System.Runtime.Serialization.Formatters" version="4.3.0" targetFramework="monoandroid71" />
```

```
<package id="System.Runtime.Serialization.Primitives" version="4.3.0" targetFramework="monoandroid71" />
```

```
<package id="System.Xml.XmlDocument" version="4.3.0" targetFramework="monoandroid71" />
```

```
<package id="Xam.Plugin.Connectivity" version="3.1.1" targetFramework="monoandroid80" />
```

```
<package id="Xam.Plugins.DownloadManager" version="1.3.5" targetFramework="monoandroid71" />
```

```
<package id="Xamarin.Android.Support.Animated.Vector.Drawable" version="25.4.0.2" targetFramework="monoandroid71" />
```

```
<package id="Xamarin.Android.Support.Annotations" version="25.4.0.2" targetFramework="monoandroid71" />
```

```
<package id="Xamarin.Android.Support.Compat" version="25.4.0.2" targetFramework="monoandroid71" />
```

```
<package id="Xamarin.Android.Support.Core.UI" version="25.4.0.2" targetFramework="monoandroid71" />
```

```
<package id="Xamarin.Android.Support.Core.Utils" version="25.4.0.2" targetFramework="monoandroid71" />
```

```
<package id="Xamarin.Android.Support.Design" version="25.4.0.2" targetFramework="monoandroid71" />
```

```
<package id="Xamarin.Android.Support.Exif" version="25.1.0" targetFramework="monoandroid71" />
```

```
<package id="Xamarin.Android.Support.Fragment" version="25.4.0.2" targetFramework="monoandroid71" />
```

```
<package id="Xamarin.Android.Support.Media.Compat" version="25.4.0.2" targetFramework="monoandroid71" />
```

```
<package id="Xamarin.Android.Support.Transition" version="25.4.0.2" targetFramework="monoandroid71" />
```

```
<package id="Xamarin.Android.Support.v4" version="25.4.0.2" targetFramework="monoandroid71" />
```

## Android projekt:

```
<package id="Xamarin.Android.Support.v7.AppCompat" version="25.4.0.2"
targetFramework="monoandroid71" />
```

```
<package id="Xamarin.Android.Support.v7.CardView" version="25.4.0.2"
targetFramework="monoandroid71" />
```

```
<package id="Xamarin.Android.Support.v7.MediaRouter" version="25.4.0.2"
targetFramework="monoandroid71" />
```

```
<package id="Xamarin.Android.Support.v7.Palette" version="25.4.0.2"
targetFramework="monoandroid71" />
```

```
<package id="Xamarin.Android.Support.v7.RecyclerView" version="25.4.0.2"
targetFramework="monoandroid71" />
```

```
<package id="Xamarin.Android.Support.Vector.Drawable" version="25.4.0.2"
targetFramework="monoandroid71" />
```

```
<package id="Xamarin.Build.Download" version="0.4.2" targetFramework="monoandroid71" />
```

```
<package id="Xamarin.FFImageLoading" version="2.3.4" targetFramework="monoandroid71" />
```

```
<package id="Xamarin.FFImageLoading.Forms" version="2.3.4" targetFramework="monoandroid71" /
>
```

```
<package id="Xamarin.Forms" version="2.5.0.122203" targetFramework="monoandroid80" />
```

## iOS projekt

```
<package id="Acr.Support" version="2.1.0" targetFramework="xamarinios10" />
```

```
<package id="Acr.UserDialogs" version="6.4.1" targetFramework="xamarinios10" />
```

```
<package id="akavache" version="5.0.0" targetFramework="xamarinios10" />
```

```
<package id="akavache.core" version="5.0.0" targetFramework="xamarinios10" />
```

```
<package id="akavache.sqlite3" version="5.0.0" targetFramework="xamarinios10" />
```

```
<package id="BTPProgressHUD" version="1.2.0.6" targetFramework="xamarinios10" />
```

```
<package id="Microsoft.CSharp" version="4.4.1" targetFramework="xamarinios10" />
```

```
<package id="Microsoft.NETCore.Platforms" version="2.0.1" targetFramework="xamarinios10" />
```

```
<package id="Mindscape.Raygun4Net" version="5.5.4" targetFramework="xamarinios10" />
```

```
<package id="NETStandard.Library" version="2.0.1" targetFramework="xamarinios10" />
```

```
<package id="Newtonsoft.Json" version="10.0.3" targetFramework="xamarinios10" />
```

```
<package id="Rg.Plugins.Popup" version="1.0.4" targetFramework="xamarinios10" />
```

```
<package id="Rx-Core" version="2.2.5" targetFramework="xamarinios10" />
```

```
<package id="Rx-Interfaces" version="2.2.5" targetFramework="xamarinios10" />
```

```
<package id="Rx-Linq" version="2.2.5" targetFramework="xamarinios10" />
```

```
<package id="Rx-Main" version="2.2.5" targetFramework="xamarinios10" />
```

iOS projekt	
<package id="Rx-PlatformServices" version="2.2.5" targetFramework="xamarinios10" />	
<package id="Splat" version="1.6.2" targetFramework="xamarinios10" />	
<package id="SQLitePCLRaw.bundle_e_sqlite3" version="1.1.0" targetFramework="xamarinios10" />	
<package id="SQLitePCLRaw.core" version="1.1.0" targetFramework="xamarinios10" />	
<package id="SQLitePCLRaw.lib.e_sqlite3.ios_unified.static" version="1.1.0" targetFramework="xamarinios10" />	
<package id="SQLitePCLRaw.provider.internal.ios_unified" version="1.1.0" targetFramework="xamarinios10" />	
<package id="System.ComponentModel.TypeConverter" version="4.3.0" targetFramework="xamarinios10" />	
<package id="System.IO.Compression" version="4.3.0" targetFramework="xamarinios10" />	
<package id="System.IO.Compression.ZipFile" version="4.3.0" targetFramework="xamarinios10" />	
<package id="System.Reflection.TypeExtensions" version="4.3.0" targetFramework="xamarinios10" />	
<package id="System.Runtime.CompilerServices.Unsafe" version="4.4.0" targetFramework="xamarinios10" />	
<package id="System.Runtime.Serialization.Formatters" version="4.3.0" targetFramework="xamarinios10" />	
<package id="System.Runtime.Serialization.Primitives" version="4.3.0" targetFramework="xamarinios10" />	
<package id="System.Xml.XmlDocument" version="4.3.0" targetFramework="xamarinios10" />	
<package id="WebP.Touch" version="1.0.7" targetFramework="xamarinios10" />	
<package id="Xam.Plugin.Connectivity" version="3.1.1" targetFramework="xamarinios10" />	
<package id="Xam.Plugins.DownloadManager" version="1.3.5" targetFramework="xamarinios10" />	
<package id="Xamarin.FFImageLoading" version="2.3.4" targetFramework="xamarinios10" />	
<package id="Xamarin.FFImageLoading.Forms" version="2.3.4" targetFramework="xamarinios10" />	
<package id="Xamarin.Forms" version="2.5.0.122203" targetFramework="xamarinios10" />	

UWP projekt:	
<PackageReference Include="Acr.UserDialogs"><Version>6.4.1</Version></PackageReference>	
<PackageReference Include="akavache"><Version>5.0.0</Version></PackageReference>	
<PackageReference Include="akavache.core"><Version>5.0.0</Version></PackageReference>	
<PackageReference Include="akavache.sqlite3"><Version>5.0.0</Version></PackageReference>	
<PackageReference Include="Microsoft.CSharp"><Version>4.4.1</Version></PackageReference>	

UWP projekt:	
<PackageReference Include="Microsoft.NETCore.Platforms"><Version>2.0.1</Version></PackageReference>	
<PackageReference Include="Microsoft.NETCore.UniversalWindowsPlatform"><Version>6.0.6</Version></PackageReference>	
<PackageReference Include="Mindscape.Raygun4Net"><Version>5.5.4</Version></PackageReference>	
<PackageReference Include="NETStandard.Library"><Version>2.0.1</Version></PackageReference>	
<PackageReference Include="Newtonsoft.Json"><Version>10.0.3</Version></PackageReference>	
<PackageReference Include="Rg.Plugins.Popup"><Version>1.0.4</Version></PackageReference>	
<PackageReference Include="System.IO.Compression"><Version>4.3.0</Version></PackageReference>	
<PackageReference Include="System.IO.Compression.ZipFile"><Version>4.3.0</Version></PackageReference>	
<PackageReference Include="Xam.Plugin.Connectivity"><Version>3.1.1</Version></PackageReference>	
<PackageReference Include="Xam.Plugins.DownloadManager"><Version>1.3.5</Version></PackageReference>	
<PackageReference Include="Xamarin.FFImageLoading"><Version>2.3.4</Version></PackageReference>	
<PackageReference Include="Xamarin.FFImageLoading.Forms"><Version>2.3.4</Version></PackageReference>	
<PackageReference Include="Xamarin.Forms"><Version>2.5.0.122203</Version></PackageReference>	

## C. Nadogradnja i izmjene aplikacija

### I. Tehnički preduvjeti

Xamarin.Forms aplikacije mogu se pisati za sljedeće operativne sustave:

- iOS 8 ili viši,
- Android 4.0.3 (API 15) ili viši
- Windows 10 Universal Windows Platform.

Razvojni inženjer za rad na aplikaciji treba znanja i vještine za razvoj projekata .NET Standard i Shared Projects.

Na Xamarin.Forms GitHubu (<https://github.com/xamarin/Xamarin.Forms/wiki/Platform-Support>) je dostupan status za platforme Samsung Tizen, MacOS, GTK# i WPF.

Za Android je potrebno imati najnoviji Android SDK Tools i Android API platformu instaliranu.

*Update* na najnoviju verziju se može obaviti preko Android SDK Managera (<https://docs.microsoft.com/en-us/xamarin/android/get-started/installation/android-sdk>). Dodatno, **target/compile** verzija za Android projekte mora biti postavljena na "*Use latest installed platform*". Minimalna verzija može biti postavljena na API 15 kako bi se mogli podržati uređaji Android 4.0.3 ili noviji.

Postavke u Visual Studiu, pod Project Options > Application > Application Properties su:

- Compile using Android version: (Target Framework): **Android 6.0 (Marshmallow)**
- Minimum Android to target: Android 4.0.3 (API Level 15 - Ice Cream Sandwich)
- Target Android version: Android 6.0 (API Level 23 - Marshmallow)

Za Mac je potrebno imati OS X El Capitan (10.11) ili noviji. Za razvoj iOS aplikacija treba instalirati barem iOS 10 SDK i Xcode 8. Napomena: Windows aplikacije se ne mogu razvijati na macOS-u.

Za Windows je potrebno imati bilo koju verziju Windowsa koja podržava Xamarin. Minimalno Visual Studio 2017 ili noviji te Windows 7 ili viši. Networked Mac je potreban za iOS razvoj.

Na ovoj poveznici je opisano kako instalirati Xamarin <https://docs.microsoft.com/en-us/xamarin/cross-platform/get-started/installation/index>

## D. Popis aplikacija

### Biologija 1 i 2:

- Android: <https://play.google.com/store/apps/details?id=hr.carnet.dos.bio12>
- iOS: <https://itunes.apple.com/us/app/e-%C5%A1kole-biologija-1-2/id1344822197>
- UWP: <https://www.microsoft.com/store/apps/9MZJMV4XVCTR>

Aplikacije rade na smartphone i tablet uređajima, ali s prilagođenim layoutom. UWP aplikacija je univerzalna - radi na svim uređajima, čak i osobnim računalima s instalacijom Windows operativnog sustava.

## 7. Priručnici

Priručnici su u svrhu arhiviranja dostavljeni u izvornim datotekama u InDesign formatu, zajedno sa svim pripadajućim priložima.

U slučaju naknadnog doručivanja, izmjene se trebaju raditi u programu InDesign, a zatim datoteke sačuvati u formatu pdf i epub iz izvorišne InDesign datoteke.

Priručnici su dostavljeni u sljedećim formatima, s pripadajućim priložima :

1. Izvorne InDesign datoteke i prilozi - cjeloviti priručnici
2. Pdf datoteka cijelog priručnika
3. epub datoteka cijelog priručnika
4. Word datoteka cijelog priručnika
5. pdf datoteke pojedinih modula
6. epub datoteke pojedinih modula