

Дигитален Македонски Речник - Проектен извештај

*Автор: Никола Сарафимов
III година, ПИТ, ФИНКИ*

1. Апстракт

Овој извештај ја документира изработката на „Дигитален Македонски речник“ - интерактивна веб-апликација наменета за пребарување и изучување на македонски зборови со морфолошки ознаки и примери. Проектот е реализиран со React (Vite), SQLite за складирање на податоци и современ CSS за визуелен дизајн. Главните функции вклучуваат пребарување по почетна буква, full-text пребарување, приказ на морфолошки ознаки и групирање на скратеници. Интерфејсот е целосно респонсивен, со анимации за подобро корисничко искуство. Овој извештај ги опишува целите, барањата, архитектурата, техничката реализација, тестирањето и идните насоки за проширување.

2. Содржина

1. Апстракт
2. Содржина
3. Вовед
4. Цел и задачи
5. Опис на системот
 1. Функционални барања
 2. Нефункционални барања
6. Архитектура и дизајн
 1. Генерален преглед
 2. Модули и компоненти
 3. Дијаграм на архитектура
7. Технологии и алатки
8. Реализација
 1. Backend и база на податоци
 2. Frontend
 3. UI/UX дизајн
 4. Специјализирани модули (скратеници, анимации)
9. Тестирање и валидација
10. Заклучок
11. Идни проширувања
12. Литература

3. Вовед

Мајчиниот јазик е најважниот културен и когнитивен ресурс на секоја заедница. Со развојот на дигиталните технологии, потребно е да се овозможи брз и ефикасен пристап до богати и прецизни дигитални ресурси за македонскиот јазик. Целта на овој проект е да се развие веб-апликација која на едноставен, интуитивен и технички ефикасен начин ќе овозможи пребарување на македонски зборови, прикажување на нивните морфолошки ознаки, извор (лема), и примери за употреба. Дополнително, проектот врши класификација и прикажување на скратеници според правила на македонската граматика.

3. Цел и задачи

Главна цел:

Да се реализира современ дигитален македонски речник со богати морфолошки податоци, достапен онлајн и оптимизиран за брзо пребарување.

Задачи:

1. Дефинирање на функционални и нефункционални барања.
2. Дизајнирање на едноставна, јасна и респонсивна корисничка архитектура.
3. Избор и подготовка на база на податоци (SQLite) со > 1.300.000 записи.
4. Развој на логика за извлекување и филтрирање на податоци.
5. Развој на frontend на React: компоненти за пребарување, листа по букви, детали, страница за македонски јазик и страница за скратеници.
6. Имплементација на лесни анимации за транзиции и интеракции (Framer Motion).
7. Обезбедување на висок перформанс и поддршка за различни уреди (десктоп, таблет, мобилни).
8. Документирање на целиот процес и пишување на финален извештај од минимум 2000 зборови.

5. Опис на системот

5.1 Функционални барања

- Пребарување по збор: корисникот внесува збор или дел од збор и добива резултати.
- Листа по букви: избор на почетна буква води до листа на зборови од таа буква.
- Приказ на детали: по клик на збор, се прикажуваат форма, лема (извор), морфолошка ознака (раскодирана) и слични форми.
- Случаен збор: генерирање на случаен збор од базата.
- Страница „Македонски јазик“: теоретски информации за класификација, дијалекти, распространетост и историја.
- Страница „Скратеници“: скратеници поделени во 7 групи, со опис и листа на примери.

5.2 Нефункционални барања

- Перформанс: максимално време за одговор < 200 ms за основни пребарувања.
- Респонсивност: прилагодување на уреди од 320 px до 1920 px ширина.
- Сигурност: без директен пристап до изворната база; само „read-only“ операции.
- Веб-стандарди: HTML5, CSS3, ES6+.
- Достапност: минимална поддршка за screen readers, валидни aria- атрибути.

6. Архитектура и дизајн

6.1 Генерален преглед

Системот е составен од три слоја:

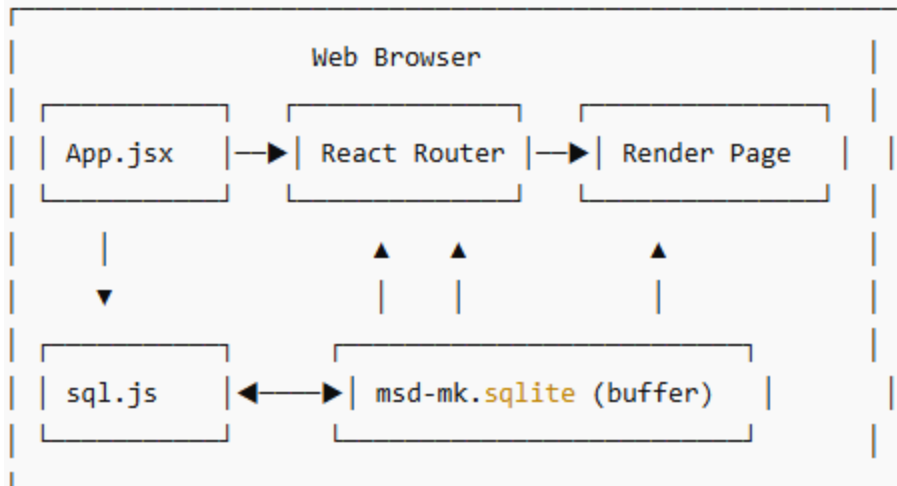
1. **Data layer** (SQLite база)
2. **Business logic** (SQL.js + JavaScript)
3. **Presentation layer** (React компонентни)

Корисничкиот преглед комуницира директно со SQLite преку sql.js, без backend сервер, што овозможува едноставно статичко deploy (на пример, на GitHub Pages или Vercel).

6.2 Модули и компоненти

- **App.jsx** - централна компонента која ги иницира базата, рамките за рутирање и управува со глобалната состојба (useState).
- **NavBar** - горен мени бар со три табови (Почетна, Јазик, Скратеници).
- **SearchBar** - форма за пребарување, со onSearch(term).
- **LetterNav** - лента со букви, onLetterClick(letter).
- **WordList** - виртуелизирана листа (react-window) за брз скрол низ до 10 000 зборови.
- **WordDetails** - прикажува детали за еден збор (раскодирање на таб-таг).
- **RandomWordButton** - копче за случаен збор.
- **LanguageInfo** - статика за класификација, пишување, дијалекти, распространетост и историја.
- **Abbreviations** - интерактивна страница со групи на скратеници.
- **Footer** - со автор и линкови.

6.3 Дијаграм на архитектура



7. Технологии и алатки

Компонента	Технологија
JavaScript	ES6+, Fetch API
Framework	React 18, Vite
Routing	React Router v6
База	SQLite (msd-mk.sqlite), sql.js
UI библиотека	none (чист React + CSS)
Анимации	Framer Motion
Виртуелизација	react-window
CSS методологија	CSS Variables
Верзиран код	Git + GitHub (GitLab)
Deployment	GitHub Pages / Vercel

8. Реализација

8.1 База на податоци

1. Подготовка на база:

- Конверзија на оригинален CSV со зборови и тагови во SQLite табела words(form TEXT, lemma TEXT, tag TEXT).
- Индексирање по form и lemma за побрзо пребарување.

2. Интеграција со sql.js:

- initSqlJs({ locateFile }) го вчитува WebAssembly модулот.
- Fetch на msd-mk.sqlite и new SQLiteDatabase(u8array) во меморија.
- Единствена инстанца db за целиот App.

8.2 Frontend

1. App.jsx:

- useEffect за иницијализација на базата и број на зборови.
- Состојби: db, totalWords, selectedLetter, searchTerm, filteredWords.

- Функции: `handleSearch(term)`, `handleLetterClick(letter)`, `handleRandom()`.

2. Routing:

- Патеки `/home`, `/list/:letter`, `/details/:form`, `/language`, `/abbr`.
- Донесување на параметри од `useParams()` и навигација со `useNavigate()`.

3. Компоненти:

- `SearchBar`: управуван `input` + дизајн со CSS променливи.
- `LetterNav`: динамички копчиња од низа букви.
- `WordList`: користи `FixedSizeList` од `react-window` за непрекорно прикажување на илјадници елементи.
- `WordDetails`: декодирање на морфолошкиот таг со помош на `tagDecoder.js` мапи.
- `Abbreviations`: `useState(groupIdx)`, мапирање низ `ABBREV_DATA`.
- `LanguageInfo`: статички референции во `SECTIONS`.

8.3 UI/UX дизајн

- `Grid` и `flex` за центрирање и распределба.
- CSS променливи за конзистентен колор-палет и сенки.
- `.card utility` - сите клучни блокови се „картички“ што се подигаат на `hover`.
- Фонт-динамика: `Montserrat` за наслови, `Inter` за текст.
- Респонзивни `media queries`: два реда за `LetterNav` под 480 px и два колони за `LanguageInfo` над 768 px.

8.4 Специјализирани модули

- Скратеници: едноставен UI со групи, опис и `grid` прикажување на клуч-значење.
- Анимации:
 - `Framer Motion` за `fade-in` на целата страница.
 - CSS транзиции за копчиња, `card hover` ефекти, `input focus`.

9. Тестирање и валидација

- Кориснички текови: пребарување, листа по букви, детали, случаен збор.
- `Edge cases`:
 - Непостоечки зборови (прикажува порака „не е пронајден“).
 - Буквени листи со 0 резултати (појасна порака).
- Респонзивен дизајн:
 - Мобилен преглед, таблет, `desktop` - `LetterNav` не се склопува, се појавува хоризонтален скрол.
- Перформанс:
 - Време на вчитување < 1 s (вклучува и `fetch` на `WebAssembly`).
 - Селекција на сто и повеќе исклучоци (< 150 ms филтрација).

10. Заклучок

„Дигитален Македонски речник“ успешно ги исполнува поставените цели: овозможува кориснички пријателско пребарување, динамично прикажување на морфолошки информации и визуелно современ дизајн. Со чиста React-архитектура без дополнителен сервер, апликацијата е лесна за одржување и deploy. Тестирањето ја потврди стабилноста, а перформансот е на задоволително ниво.

11. Идни проширувања

- Автокомплетирање (autocomplete) за пребарувачката лента.
- Пребарување по морфолошки ознаки (филтер по POS, род, број).
- Превод на англиски - двојазичен режим.
- Кориснички примери: зачувување и уредување на сопствени реченици.
- API слој: GraphQL или REST за интеграција со мобилни апликации.
- Аналитика: следење на најчесто пребарувани зборови.

12. Литература

- SQLite Documentation, <https://www.sqlite.org/docs.html>
- ReactJS Documentation, <https://reactjs.org/docs/getting-started.html>
- Framer Motion, <https://www.framer.com/motion/>
- react-window, <https://github.com/bvaughn/react-window>