Generiranje dinamičkih korisničkih obrazaca pomoću programskog okvira vue.js

Cunj, Marko

Undergraduate thesis / Završni rad

2021

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Pula / Sveučilište Jurja Dobrile u Puli**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:137:754800

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2025-03-28**

Repository / Repozitorij:

Digital Repository Juraj Dobrila University of Pula

Sveučilište Jurja Dobrile u Puli

Fakultet informatike u Puli

MARKO CUNJ

GENERIRANJE DINAMIČKIH KORISNIČKIH OBRAZACA POMOĆU PROGRAMSKOG OKVIRA VUE.JS

Završni rad

Pula, svibnja 2021.

Sveučilište Jurja Dobrile u Puli

Fakultet informatike u Puli

MARKO CUNJ

GENERIRANJE DINAMIČKIH KORISNIČKIH OBRAZACA POMOĆU PROGRAMSKOG OKVIRA VUE.JS

Završni rad

JMBAG: 0303085565, redovni student

Studijski smjer: Informatika

Kolegij: Programsko inženjerstvo

Znanstveno područje: Društvene znanosti

Mentor: Doc. dr. sc. Nikola Tanković

Pula, svibnja 2021.

Sadržaj

1. Uvod 1

2. Motivacija 2

3. Vue Form Builder 3

3.1. Općenito o Vue Form Builderu 3

3.2. Zašto Vue? 4

3.3. „Form Builder“ 5

3.3. „Form Renderer“ 6

3.5. „Vue-multiselect“ 8

3.6. Vue2-dropzone 10

4. Paketi korišteni u projektu 11

Zaključak 12

Sažetak 13

Literatura 14

Popis slika 15

1. Uvod

Primjena korisničkih obrazaca u današnje vrijeme predstavlja izazov u trošenju vremena kao i novaca na kreiranje i pisanje koda za svaku formu koju programer primijeni u svoj kod, pogotovo ako je potrebno stvoriti veliki korisnički obrazac u kojem su ulazni podaci međusobno slični i često se mijenjaju. Ovdje na scenu stupa generiranje dinamičkih korisničkih obrazaca, što je ujedno i tema ovog rada.

Ovaj rad se temelji na projektu kojeg je razvio programer iz Vijetnama pod nazivom „Vue Form Builder“ (hrv. Vue generator korisničkih obrazaca). Cilj ovog rada je proširiti projekt dodajući mu više funkcionalnosti, ujedno i maknuti one manje potrebne. Također, u projekt je uključen i uklopljen „Vue Multiselect“, komponenta koja služi kao mogućnost automatskog popunjavanja polja. Omogućeno je više odabira unutar jednog polja. „Vue Form Builder“ i „Vue Multiselect“ su uklopljeni na način koji najviše odgovara BPMN specijaliziranom računalu koji će biti opisan u nastavku.

„Vue Form Builder“ komponenta je, odnosno dodatak, koja pomaže krajnjem korisniku (programeru) kreirati jednostavnu formu s dodavanjem, „primi & pusti“ itd., da bi naposljetku dobio kod u JSON formatu kojim se kasnije pomoću „Form Renderer“ dijela projekta učitava bez tipkanja koda. „Vue Form Builder“ napravljen je potpuno u Vue okruženju.

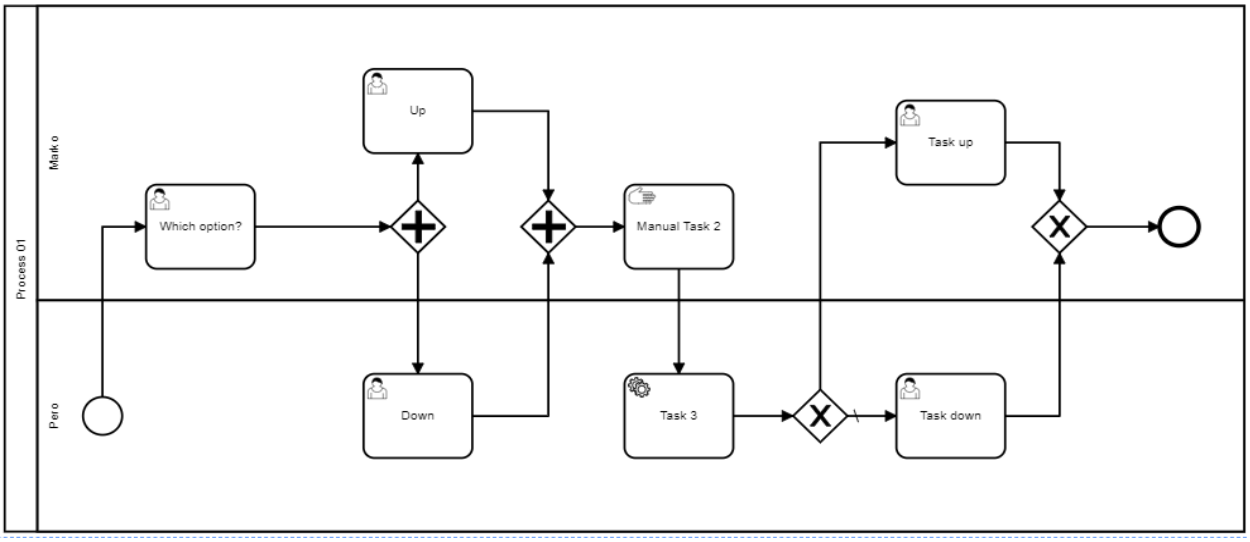
„GitHub“ hiperveza repozitorija glasi: „https://github.com/markocunj/vue-form-builder“.

„GitHub“ hiperveza testnog repozitorija glasi: „https://github.com/markocunj/testniGit“.

2. Motivacija

Motivacija za izradu ovog završnog rada je to što će služiti u jednom BPMN specijaliziranom računalu kao glavni segment dizajna i prevođenja korisničkih obrazaca.

BPMN (engl. Business Process Modeling Notation) je vizualni jezik za poslovnu analizu i specificiranje tijekova procesa u određenom poduzeću. BPMN je popularna i intuitivna grafika koja se može lako shvatiti od strane programera, analitičara, arhitekta informacijama itd.



Slika 1. Primjer BPMN procesa

Prema primjeru BPMN procesa, ikone korisniku omogućavaju lakše razumijevanje procesa. Cijeli proces prati logički slijed događaja, svi koraci trebaju biti odrađeni kako bi cjelokupni BPMN proces bio završen.

Ovaj rad u BPMN-u je dvojak: dio koji ispunjava korisnik, ali i dio procesa koje mora riješiti stroj (računalo). Naime, računalo pred korisnika stavlja obrazac koji je dužan ispuniti. Računalo, nakon što korisnik preda podatke, mora procesirati korisnički obrazac i poslati ga na određeno mjesto.

3. Vue Form Builder

U ovom poglavlju će biti detaljno objašnjene funkcionalnosti ovoga projekta, kao što su dijelovi „Form Builder“ i „Form Renderer“ od kojega se sastoji sam „Vue Form Builder“, kao i prednosti Vue programskog okvira.

3.1. Općenito o Vue Form Builderu

Vue Form Builder je vrlo koristan jer programerima smanjuje vrijeme kreiranja određenih formi. Vue Form Builder se sastoji od dvije glavne komponente koje se kasnije dijele na puno malih.

Te dvije glavne komponente su:

- „Form Builder“ komponenta
- „Form Renderer“ komponenta

Glavne prednosti ovog projekta su:

- jednostavan UI/UX koji pomaže programeru u izradi njegove forme
- manje koda u programiranju
- ušteda vremena programeru
- mogućnost automatskog popunjavanja polja koja ima mogućnost dohvaćanja podataka s „pozadine“ (engl. back-end) uz pomoć „Axiosa“ (po želji programera)
- jednostavna instalacija
- jednostavna implementacija (demonstrirano u nastavku teksta)

„Form Builder“ je dio projekta koji služi za stvaranje same forme što uključuje dodavanje sekcija, kontrola, konfiguracija, dodavanje naslova, podnaslova itd. Nakon toga se iz njega dobiva JSON shema koju programer ubacuje u varijablu „formData“ u svojem kodu kako bi komponenta mogla prepoznati što mora učitati.

„Form Renderer“ je dio koji prikazuje ono što je odabrano u „Form Builderu“. „Form Renderer“ u sklopu s generiranom JSON shemom strukture forme iz prošlog dijela učitava formu i dozvoljava krajnjem korisniku da ju koristi.

Instalacija ovog projekta (preko NPM-a) je veoma jednostavna i izgleda ovako:

„npm install https://github.com/markocunj/vue-form-builder“

Nakon što je paket ubačen u projekt, potrebno ga je aktivirati u glavnoj datoteci JavaScript dijela projekta (npr. „main.js“).

```
import { VueFormBuilderPlugin } from "v-form-builder";
import "v-form-builder/dist/v-form-builder.css"; //Dodaje CSS vašem projektu
// install now
Vue.use(VueFormBuilderPlugin);
```

Slika 2. Prikaz uvoza projekta u kod

3.2. Zašto Vue?

Vue je trenutno jedan od glavnih tehnologija „prednje-strane“ (engl. front-end) te je sličan tehnologijama poput „Angulara“ i „React.js-a“.

Glavne prednosti Vue-a su:

- prilagođenost korisnicima – jako je koristan novim programerima zbog toga što ne zahtijeva od programera poznavanje više od osnova JavaScripta, HTML-a i CSS-a, za razliku od „Angulara“ ili „Reacta“ koji od programera zahtijevaju znanje dodatnih programskih jezika za napredno kodiranje. Vue čak ima i dodatak na pretraživačima Chrome i Firefox gdje se novim korisnicima pomaže što jednostavnije shvaćanje potrebnoga.
- prilagodba – nudi prilagodbu individualnim zahtjevima krajnjih korisnika. Naime, programer može nazvati funkciju po svojoj želji i svaki od segmenata aplikacije može imati odvojene funkcije.
- malo ograničenja – dizajn Vue-a nudi manje ograničenja i više fleksibilnosti kako bi se projekt završio. Središte samog okruženja se temelji na „View“ dijelu koji u kombinaciji s modularnim pristupom i korištenjem različitih dostupnih paketa dozvoljava programerima rješavati probleme različitim načinima.
- jednostavnost – glavna ideja razvoja Vue-a je postizanje dobrih rezultata sa što manje uloženog truda tako da korisnik može kodirati koristeći samo nekoliko linija koda.

- integracija – programerima je omogućeno integrirati Vue u druge programske okvire kao što je „React“ i omogućiti prilagodbu projekta prema njihovim zahtjevima.

Zbog navedenih mogućnosti, Vue je odličan odabir programskog okvira koji služi razvijanju ovog rada. Krivulja korištenosti Vue-a u odnosu na „React“ i „Angular“ je u uzlaznoj putanji.

3.3. „Form Builder“

Ova komponenta je dio „Vue Form Buildera“. Sastoji se od sekcija, redova, kontrola, specifikacija kontrola itd. te je samim time lako dostupan korisnicima (engl. „user-friendly“), korisnik neće imati problema sa shvaćanjem uporabe „Buildera“, ali kako bi mogao koristiti „Builder“, potrebno je prvo ga uključiti u svoj projekt (nakon što ga je instalirao kao što je pokazano u poglavlju 3.1.)

Uključivanje „FormBuildera“:

```
<template>
  <FormBuilder></FormBuilder>
</template>

<script>
import { FormBuilder } from "v-form-builder";

export default {
  components: { FormBuilder },
  data: () => ({}),
};
</script>
```

Slika 3. Uključivanje "Form Builder" objekta u projekt

Ovo je vrlo jednostavan postupak. Tek nekoliko linija teksta potrebno je kako bi „Form Builder“ bio uključen u projekt (pod uvjetom da je inicijaliziran u „main.js-u“).

Korisnički obrazac je jednostavno napraviti. U ovome radu neće biti posebnog osvrta na detalje izrade korisničkog obrasca jer je individualan za svakog korisnika.

Nakon stvaranja korisničkog obrasca, potrebno je stisnuti gumb „Get-JSON FormData“ koji potom ispisuje konfiguraciju forme u korisničku konzolu iz koje se kopira JSON tekst koji će biti potreban za učitavanje forme.

3.4. „Form Renderer“

„Form Renderer“ je drugi, vrlo važan, dio ovog projekta bez kojega korisnik ne bi mogao vidjeti ni koristiti se korisničkim obrascem koji je napravio u prvom dijelu. Kao i kod kreiranja obrasca, potrebno je uvesti određene objekte u kod. Kada se podaci dobiveni iz generiranja forme (JSON shema), unesu u „Form Renderer“, on ih učitava, odnosno stvara korisnički obrazac koji je potreban i koji je izabran u kreiranju forme.

Uključivanje „Form Renderera“:

- „<template>“ dio:

```
<template>
    <FormRenderer
            :form-configuration="formData"
            v-model="formInputData"
    />
</template>
```

Slika 4. Uključivanje "Form Renderera" u kod (1/2)

- „<script>“ dio

```
<script>
    import {FormRenderer} from 'v-form-builder'

    export default {
        components: {
            FormRenderer,
        },

        data() {
            return {
                formData: null,
                formInputData: null,
            }
        },

        methods: {
            getFormData() {
                // get form data...

                //...
                this.formData = ////;
            }
        },

        created() {
            this.getFormData()
        }
    }
</script>
```

Slika 5. Uključivanje "Form Renderera" u kod (2/2)

U „script“ dijelu programer, odnosno korisnik, unosi konfiguraciju koju je dobio kod kreiranja obrasca u varijablu „formData“. „V-model=“formInputData““ daje programeru povratnu informaciju u obliku podataka koje je krajnji korisnik upisao u formu i prema tome programer raspolaže tim podacima i sprema ih u bazu podataka (ili po potrebi nešto drugo).

3.5. „Vue-multiselect“

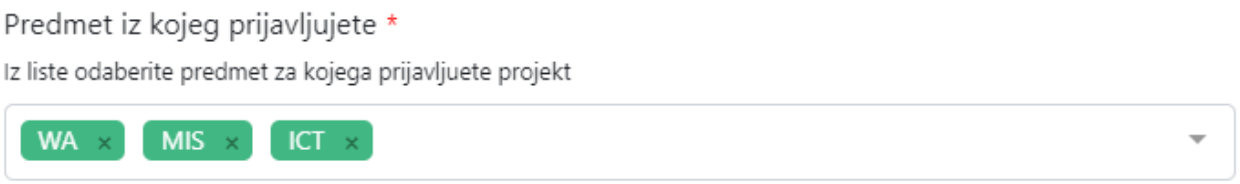
„Vue-multiselect“ je paket koji je uklopljen posebno unutar ovog završnog rada. „Vue-multiselect“ je koristan zbog toga što omogućuje puno značajki koje su potrebne za automatsko popunjavanje polja u ovom projektu.

Neke od značajki „Vue-multiselecta“ su:

- jedan / višestruki odabir
- padajući izbornik (engl. dropdowns)
- mogućnost dohvaćanja podataka iz „pozadine“ (engl. back-end)

„Vue-multiselect“ nije potrebno posebno provoditi zbog toga što je on već proveden i nalazi se u „package.json“ datoteci projekta.

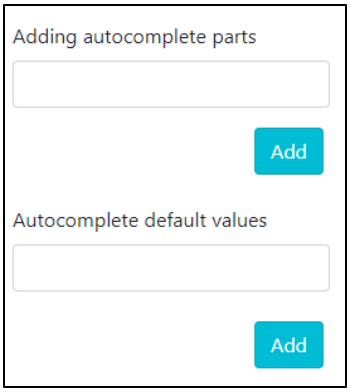
Primjer višestrukog odabira u „Vue-multiselect“:



Slika 6. Prikaz višestrukog odabira

Unutar opcija odabira korisniku je omogućen izbor između umetanja dijelova za automatsko ručno popunjavanje polja ili upisivanje hiperveze koja vodi do polja koje se potom uključuje u kod. Također, postoji mogućnost kojom korisnik može odabrati višestruki unos.

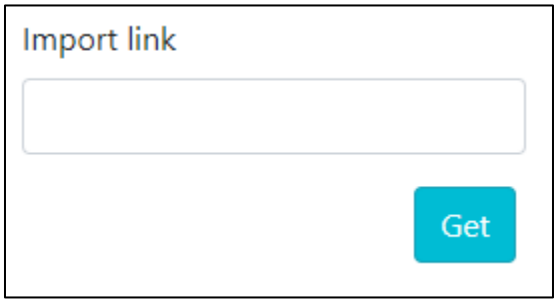
U nastavku rada nalazi se primjer izgleda mogućnosti korisničkog popunjavanja :



Slika 7. Izgled unosa korisnika

U ovom dijelu korisnik pod opcijom „Adding autocomplete parts“ dodaje dijelove koje želi vidjeti u konačnom izgledu forme. Ispod toga se nalazi polje „Autocomplete default values“ u koje korisnik ubacuje vrijednosti koje želi imati upisane kad se forma otvori.

Primjer u nastavku prikazuje polje „Import link“ u koji korisnik, ako se odlučio na dohvaćanje podataka s hiperveze, mora unijeti hipervezu koja vodi do stranice na kojoj se nalazi polje koje će se koristiti u konačnoj verziji forme.

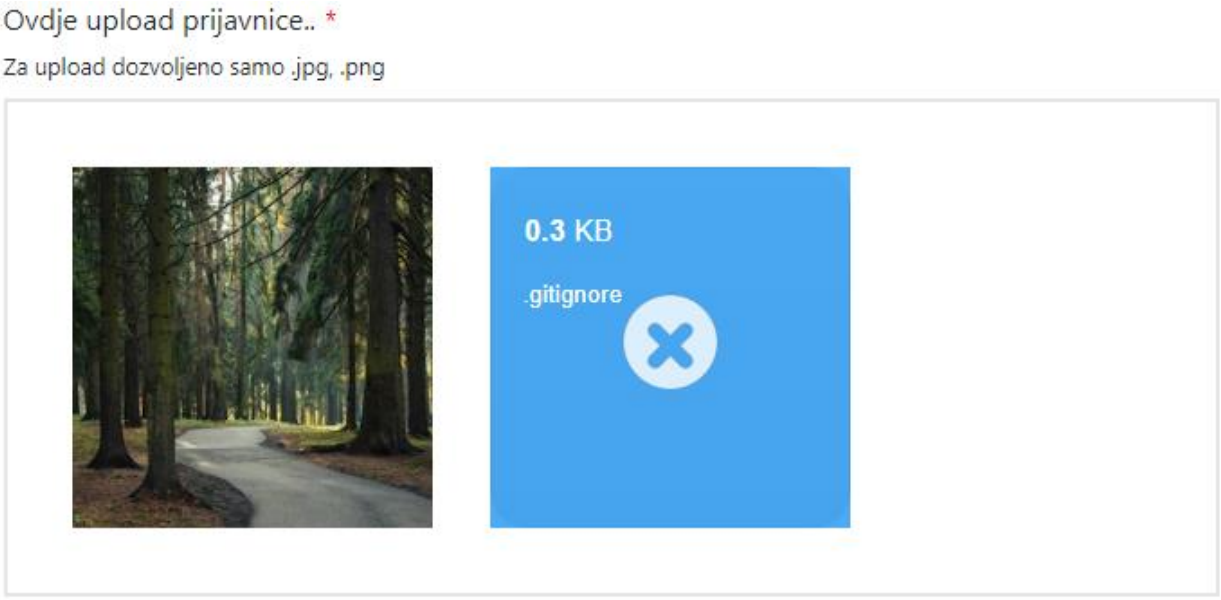


Slika 8. Prikaz unosa putem hiperveze

Korisniku će se nakon uspješnog dohvaćanja podataka prikazati poruka kako su podaci dohvaćeni, u suprotnome će se pokazati poruka s greškom.

3.6. Vue2-dropzone

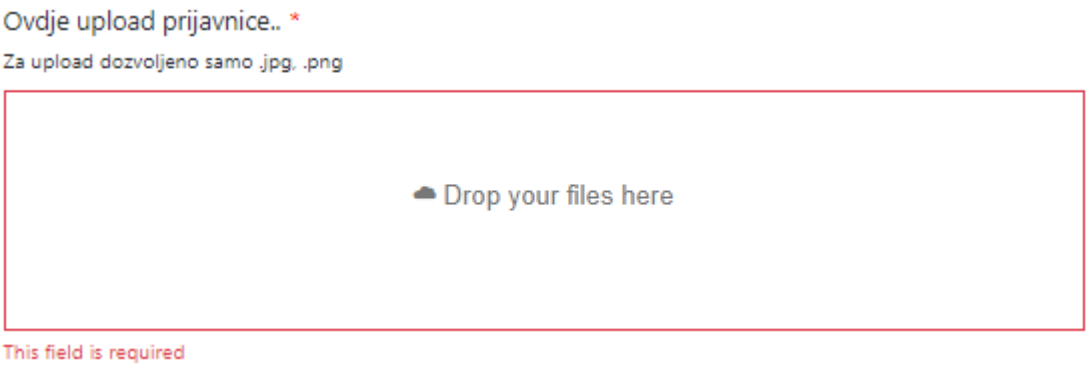
Vue2-dropzone je komponenta koja je uključena u ovaj rad kao zamjena za „Vue-upload-component“. Vue2-dropzone je inačica „dropzone.js“ paketa koji je razvijen za „JavaScript“ programski jezik, ali za Vue okruženje.



Slika 9. Primjer uspješno i neuspješno učitane datoteke

Uspješno učitana datoteka se nalazi na lijevoj strani te ona zadovoljava kriterije koji su napisani u opisu, za razliku od drugog dokumenta koji ima nedozvoljenu ekstenziju te samim time nije prihvaćen u učitavanje datoteke.

Validacija na ovoj komponenti izgleda ovako:



Slika 10. Prikaz validacije polja

4. Paketi korišteni u projektu

Bez paketa ovaj projekt bi bio teško provediv. Niže u tekstu su navedeni glavni paketi koji su povezani sa „Vue Form Builder“ i s preradama projekta.

Paketi:

- „Axios“ – služi za izradu HTTP zahtjeva iz „node.js“ i igra vitalnu ulogu u dohvaćanju podataka iz „pozadine“ (engl. back-end) kao i slanja tih podataka natrag na „pozadinu“
- „Core-js „– paket za „JavaScript“ koji uključuje „obećanja“ (engl. promise), simbole, kolekcije, iteratore (engl. iterators), polja (engl. arrays) i mnoge ostale funkcionalnosti
- „Dayjs“ – služi za prepoznavanje, validaciju, manipuliranje i prikaz datuma i vremena za moderne pretraživače poput „Google Chromea“ itd., radi u skladu s „Moment.js“
- „Litepicker“ – služi za prikazivanje datuma, odnosno raspona datuma (nekoliko mjeseci, jedan datum ili raspon datuma), moguć višestruk odabir itd.
- „Vue-axios“ – predstavlja „Axios“ dodatak, ali za Vue okruženje, pomaže pri integraciji „Axios“ u Vue
- „Vue-class-component“ – paket koji dopušta programeru raditi svoje Vue komponente u stil (engl. style) sintaksi
- „Vue-multiselect“ – opisan u poglavlju 3.5. služi kao mogućnost automatskog popunjavanja polja, metodu koja je uključena u ovaj projekt
- „Vue-property-decorator“ – paket koji ovisi o „Vue-class-component“ paketu te popunjava neke njegove nedostatke
- „Vue2-dropzone“ – komponenta koja omogućuje „primi i pusti“ opciju (eng. „drag & drop“) za odabir datoteka koje korisnik želi učitati na određenu stranicu. U poglavlju 3.6. su detaljnije objašnjeni dijelovi ove komponente i njena uključenost u projekt.
- „Vuedraggable“ – omogućuje „primi i pusti“ opciju (eng. „drag & drop“) i sinkronizaciju s poljem u kojem su spremljene kontrole.

Zaključak

U ovom završnom radu analiziran je projekt „Vue Form Buildera“. Objašnjene su osnove BPMN-a, kao i samih detalja o „Vue Form Builderu“ i generiranju dinamičkih formi unutar Vue programskog okvira. Također, objašnjene su pozitivne strane Vue okruženja, kao i neki od razloga zbog kojeg je ovaj rad (projekt) stvaran u tom okruženju, a ne u sličnim programskim okruženjima, kao što su „React“ ili „Angular“.

Projekt se sastoji od dvije glavne komponente, a to su: „FormBuilder“ komponenta i „FormRenderer“ komponenta.

„FormBuilder“ komponenta programeru/korisniku omogućuje odabir izgleda korisničkog obrasca. Ima širok raspon kontrola, od unosa preko vremenskog unosa, gumbiju i još mnogo izbora, sve do učitavanja datoteka. Osnovne značajke „FormBuildera“ su: sekcije, redovi, kontrole, validacija itd.

„FormRenderer“ komponenta programeru/korisniku omogućuje učitavanje forme koju je napravio unutar „FormBuildera“. Drugim riječima, „FormRenderer“ je prevoditelj „FormBuilder“ koda. Bez „FormRenderera“ je nemoguće učitavanje korisničkog obrasca.

U projektu je uključena i validacija koju korisnik odabire pri odabiru unosa, a neke od dostupnih validacija su: „Required field“, „E-mail check“, „Regex check“, „min“, „max“, „sameAs“ itd.

Važan dio projekta je i „Vue-multiselect“. „Vue-multiselect“ krajnjem korisniku (ako programer tako zada u „FormBuilderu“) može služiti kao mogućnost automatskog popunjavanja polja. Korisnik dobije već ponuđene vrijednosti od kojih sam odabire koja mu odgovara.

Od programskih paketa su u 4. poglavlju navedeni svi koji su korišteni u ovom projektu, a neki od najbitnijih su: „Axios“, „Vue-Axios“, „Vue-multiselect“, „Vue-class-component“, „Vue-upload-component“ i „Vue-draggable“.

Sažetak

Primjena i izrada korisničkih obrazaca predstavlja prilično veliki izazov u nedostatku vremena, ali i novaca za izradu korisničkog obrasca. Ovaj projekt programera iz Vijetnama s nadogradnjama uvelike olakšava proces izrade tih korisničkih obrazaca, kao i raspolaganje podacima koje krajnji korisnik upisuje, a programer sprema u bazu podataka ili koristi na način koji mu je potreban. Uporaba ovog projekta je relativno jednostavna. Samo upoznavanje s funkcijama je maksimalno olakšano. Svatko tko se i malo razumije u Vue okruženje, s osnovnim znanjem, može se koristiti ovim projektom i svim njegovim funkcionalnostima.

Literatura

BPMN-activity-types. (n.d.). Preuzeto 1. 6. 2021. iz Visual-paradigm: https://www.visual-paradigm.com/guide/bpmn/bpmn-activity-types-explained/

Nguyen, Q.-A. (n.d.). *Vue-axios*. Preuzeto 1. 6. 2021. iz npmjs.com: https://www.npmjs.com/package/vue-axios

Saburido, A. (n.d.). *Vue Dynamic Forms*. Preuzeto 31. 5. 2021. iz alvarosaburido.dev: https://vue-dynamic-forms.alvarosaburido.dev/

Tran, P. (n.d.). *Vue Form Builder*. Preuzeto 31. 5. 2021. iz gitbook.io: https://phattranminh96.gitbook.io/vue-form-builder/

Vasiutynska, I. (21. 3 2019). *Pros and cons of using Vue-js*. Preuzeto 31. 5. 2021. iz towardsdatascience.com: https://towardsdatascience.com/what-are-the-pros-and-cons-of-using-vue-js-3689d00d87b0

Vue-mutliselect. (n.d.). Preuzeto 1. 6. 2021. iz vue-multiselect.js.org: https://vue-multiselect.js.org/

What is BPMN? (n.d.). Preuzeto 1. 6. 2021. iz Visual-paradigm: https://www.visual-paradigm.com/guide/bpmn/what-is-bpmn/

Yue, L. (n.d.). *Vue-upload-component*. Preuzeto 31. 5. 2021. iz npmjs.com: https://www.npmjs.com/package/vue-upload-component

Zabriskie, M. (n.d.). *Axios*. Preuzeto 1. 6. 2021. iz github.com: https://github.com/axios/axios

Popis slika

Slika 1. Primjer BPMN procesa 2
Slika 2. Prikaz uvoza projekta u kod 4
Slika 3. Uključivanje "Form Builder" objekta u projekt 5
Slika 4. Uključivanje "Form Renderera" u kod (1/2) 6
Slika 5. Uključivanje "Form Renderera" u kod (2/2) 7
Slika 6. Prikaz višestrukog odabira 8
Slika 7. Izgled unosa korisnika 9
Slika 8. Prikaz unosa putem hiperveze 9
Slika 9. Primjer uspješno i neuspješno učitane datoteke 10
Slika 10. Prikaz validacije polja 10