

Web-aplikacija za praćenje korisničkih prijava neispravnosti temeljena na tehnologiji React

Petrović, Mirta

Undergraduate thesis / Završni rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Electrical Engineering and Computing / Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva**

Permanent link / Trajna poveznica: <https://urn.nsk.hr/urn:nbn:hr:168:339499>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2025-03-20**



Repository / Repozitorij:

[FER Repository - University of Zagreb Faculty of Electrical Engineering and Computing repository](#)



SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1278

**WEB-APLIKACIJA ZA PRAĆENJE KORISNIČKIH PRIJAVA
NEISPRAVNOSTI TEMELJENA NA TEHNOLOGIJI REACT**

Mirta Petrović

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1278

**WEB-APLIKACIJA ZA PRAĆENJE KORISNIČKIH PRIJAVA
NEISPRAVNOSTI TEMELJENA NA TEHNOLOGIJI REACT**

Mirta Petrović

Zagreb, lipanj 2024.

ZAVRŠNI ZADATAK br. 1278

Pristupnica: **Mirta Petrović (0036535633)**
Studij: Elektrotehnika i informacijska tehnologija i Računarstvo
Modul: Računarstvo
Mentorica: prof. dr. sc. Željka Car

Zadatak: **Web-aplikacija za praćenje korisničkih prijava neispravnosti temeljena na tehnologiji React**

Opis zadatka:

Razvoj i isporuka mobilnih aplikacija imaju određene specifičnosti, pogotovo vezano uz različite tehnološke značajke korisničkih uređaja, na različite verzije operacijskog sustava koje korisnici mogu koristiti, kao i na učestala ažuriranja mobilnih operacijskih sustava. Stoga je nužno uspostaviti mehanizam komunikacije s korisnicima u kojem bi oni mogli dojaviti uočene neispravnosti ili teškoće u radu, a kako bi se olakšalo održavanje mobilnih aplikacija. Vaš je zadatak osmisliti i u tehnologiji React razviti web-aplikaciju za korisničku prijavu neispravnosti za skupinu mobilnih aplikacija koje se koriste za komunikaciju i edukaciju. Analizirajte postojeća slična rješenja, kao i potrebe korisnika ove skupine aplikacija te na temelju toga definirajte programske zahtjeve Vašeg rješenja.

Rok za predaju rada: 14. lipnja 2024.

Sadržaj

Uvod	1
1. Web-obraci.....	3
1.1. Analiza postojećih web-obrazaca	4
2. Potrebe korisnika	8
3. Opis razvijene web-aplikacije za praćenje korisničkih prijava neispravnosti.....	10
4. Tehničke značajke razvijene web-aplikacije	12
4.1. Tehnologije i alati.....	12
4.2. Arhitektura razvijene web-aplikacije.....	13
4.3. Funkcionalnosti razvijene web-aplikacije	15
4.4. Elementi razvijene web-aplikacije.....	16
4.5. Implementacija razvijene web-aplikacije	19
5. Testiranje razvijene web-aplikacije	23
6. Primjer korištenja web-aplikacije.....	24
7. Potrebe održavanja	26
Zaključak	29
Literatura	30
Sažetak.....	32
Summary.....	33
Skraćenice.....	34

Uvod

Razvoj i isporuka aplikacija imaju određene specifičnosti vezane uz različite tehnološke značajke uređaja te različite verzije operacijskih sustava. Zbog toga je bitno održavati aplikacije i uspostaviti način komunikacije s korisnicima kako bi oni mogli dojaviti neispravnosti u aplikacijama. Jedan od načina uspostave komunikacije s korisnicima jest putem web-aplikacija.

ICT-AAC aplikacije namijenjene su komunikaciji i edukaciji. Aplikacije su razvijene u okviru kompetencijske mreže ICT-AAC. Usmjerene su na pomoć osobama sa složenim komunikacijskim potrebama u komunikaciji, edukaciji i društvenom uključivanju [1]. Razvijene su aplikacije za Apple iOS, Google Andorid i Web [2]. Zbog učestalih ažuriranja sustava i pojave novih uređaja potrebno je održavati ICT-AAC aplikacije.

Stoga je u ovome radu u tehnologiji *React* razvijena web-aplikacija za praćenje korisničkih prijava neispravnosti za skupinu mobilnih aplikacija koje se koriste za komunikaciju i edukaciju. Jotform prijava grešaka na ICT-AAC aplikacijama [3] prethodno je rješenje koje služi za prijavu grešaka pronađenih u ICT-AAC aplikacijama te se nalazi na stranici ICT-AAC [4]. Unaprijeđen je taj web-obrazac i osmišljena je metoda spremanja podataka poslanih web-obrazaca kako bi ih se kasnije moglo prikazati.

Web-aplikacija omogućuje ispunjavanje web-obrasca za prijavu pronađene neispravnosti u aplikacijama za komunikaciju i učenje te pregled poslanih prijava kako bi se neispravnosti mogle ispraviti. Glavni je cilj ovoga rada analizirati prethodna slična rješenja i potrebe korisnika, a zatim definirati programske zahtjeve web-aplikacije i naposljetku izraditi web-aplikaciju. Ciljna skupina korisnika su roditelji, pedagozi, logopedi i sve osobe koje rade i pomažu djeci u učenju i korištenju aplikacija za komunikaciju i edukaciju. Putem ove web-aplikacije, cilj je omogućiti korisnicima da lako i jednostavno mogu prijaviti grešku na koju su naišli. Jedne od glavnih funkcionalnosti web-aplikacije su omogućiti unos podataka u web-obrazac i slanje web-obrasca, spremanje poslanih podataka, čitanje i prikaz spremljenih podataka. Tako će osobe koje su zadužene za ispravljanje grešaka u što kraćem vremenskom roku moći ispraviti grešku i na taj način omogućiti djeci da nastave koristiti web-aplikaciju.

Ovaj je rad strukturiran na sljedeći način; u prvom poglavlju analizirani su postojeći web-obrasci te uspoređene su njihove prednosti i nedostatke. U sljedećem su poglavlju opisane potrebe korisnika web-aplikacije. Zatim je opisan razvijeni proizvod i zahtjevi web-aplikacije. U dijelu naslovljenom „Tehničke značajke” nabrojane su tehnologije i alati koji su korišteni tijekom razvoja web-aplikacije, opisana je arhitektura sustava i navedene su funkcionalnosti. Također su navedeni elementi web-aplikacije te je opisan i proces implementacije. Zatim je objašnjeno kako je testirana web-aplikacija. Nakon toga je opisano kako se web-aplikacija koristi i koje su potrebe održavanja. Naposljetku, u zaključku su iznijeta postignuća i doprinosi ovog rada te su elaborirane smjernice za budući razvoj web-aplikacije.

1. Web-obrasci

Web-obrasci su interaktivni elementi na web stranicama koji korisnicima omogućuju slanje podataka. Služe za prikupljanje informacija od korisnika kako bi se te informacije mogle obraditi. Postoje različite vrste web-obrazaca kao što su kontakt obrazac, obrazac za registraciju, ankete, obrasci za narudžbe, obrasci za prijavu, itd. [5] [6].

Web-obrasci korisnicima omogućavaju komunikaciju s različitim organizacijama, tvrtkama i institucijama tako što korisnici mogu podnijeti zahtjev bez potrebe za slanjem elektroničke pošte ili telefonskog poziva. Štoviše, web-obrasci olakšavaju komunikaciju jer je korisnicima jednostavnije ispuniti web-obrazac u kojem su već formulirana pitanja. Alternativa bi bila da sami moraju osmisliti poruku i poslati ju elektroničkom poštom ili pak odvojiti vrijeme bez distrakcija za telefonski poziv i to za vrijeme radnog vremena agenta za korisničku podršku. Komunikacija putem web-obrazaca je stoga korisnicima jednostavnija i brža. S druge strane, web-obrasci organizacijama omogućuju da lakše prikupljaju podatke i informacije te ih onda sortiraju i obrađuju. Također omogućuju bržu obradu informacija jer se podaci mogu automatski sortirati i prosljeđivati određenim osobama koje će ih obraditi. Na taj se način uštedi vrijeme koje bi se inače potrošilo na sortiranje web-obrazaca. Umjesto sortiranja to se vrijeme može potrošiti na obradu i rješavanje problema ili upita poslanih putem web-obrazaca. Svaka organizacija pri izradi svog web-obrasca stvara svoja pitanja koja su specifična za njihove potrebe te odlučuje izgled samog web-obrasca. Još jedna prednost je mogućnost analize podataka. Analizom se mogu uočiti neke sličnosti među svim obrascima te otkriti uzorci problema i poraditi na njima kako bi se pružilo najbolje iskustvo korisnicima. Web-obrasci su jednostavni za korištenje i lako se mogu prilagoditi potrebama korisnika i organizacija.

Stoga će se u radu prvo proučiti postojeća rješenja u web-obrascima kako bi se pojasnilo kako se organizacije povezuju s korisnicima i na koji se način može poboljšati taj proces.

1.1. Analiza postojećih web-obrazaca

Analizirana su postojeća rješenja za slanje podataka putem web-obrazaca pronađenih na različitim web stranicama. Cilj je analizom uočiti prednosti i nedostatke svakog obrasca te na kraju pronaći potencijalno bolja rješenja.

Analiziran je veći broj web-obrazaca kako bi se obuhvatilo što više različitih rješenja. Web-obrasci koji su analizirani koriste se za prijavu grešaka, unos podataka, slanje upita itd. Neki od njih su: Jotform [7], ZET [8], Fina.hr kontakti [10], Hrti [11], e-Građani [9] i Zavod za zaštitu okoliša i prirode [15]. Odabrani su ovi web-obrasce jer su na hrvatskome jeziku, a u okviru ovog rada trebalo je razviti web-obrazac koji je namijenjen primarno za hrvatsko tržište.

Analiziran je i Jotform [7] koji je na engleskom jeziku jer je to alat za razvoj web-obrazaca koji sadrži vlastiti web-obrazac. Alatom Jotform razvijeno je prethodno rješenje Jotform prijava grešaka na ICT-AAC aplikacijama [3] na kojemu se temelji jedan dio ovoga rada.

Prednosti i nedostaci analiziranih web-obrazaca prikazani su u tablici (Tablica 1.1), a u nastavku će se analizirati podatci i predstaviti rezultati.

Tablica 1.1 Usporedba navedenih postojećih web-obrasca

Web-obrazac	Prednosti	Nedostaci
Jotform prijava grešaka na ICT-AAC aplikacijama [3]	Obavezna polja Mogućnost prilaganja datoteke	
Jotform [7]	Obavezna polja Mogućnost prilaganja datoteke Mogućnost prilaganja poveznice	
ZET [8]	Obavezna polja CAPTCHA	Polje poruka nije obavezno
e-Građani [9]	Ponuđene opcije odgovora	Obavezno je samo polje za

	Obavezno polje	prijedlog ili komentar Teško uočljiva pozicija web-obrasca na web stranici
Fina.hr kontakti [10]	Obavezna polja	Polje poruka nije obavezno
Hrti [11]	Mogućnost odabira vrste pitanja Obavezna polja	
Tisak paket [12]	Obavezna polja	Teško uočljiva pozicija web-obrasca na web stranici
Konzum [13]	Mogućnost odabira kategorije pitanja Obavezna polja Mogućnost prilaganja datoteke reCAPTCHA	
Brod-plin [14]	Ponuđeni odgovori Strelica za povećanje i smanjenje brojčane vrijednosti Obavezna polja Mogućnost prilaganja datoteke	Strelica za povećanje i smanjenje brojke jako je mala
Zavod za zaštitu okoliša i prirode [15]	Obavezna polja Nalaženje trenutnog mjesta pomoću lokacije Ponuđen kalendar za unos datuma i vremena pronađene	Puno pitanja

	životinje	
	Mogućnost odabira opcija za više pitanja	

Uočene prednosti u web-obrascima su obavezna polja za unos ili odabir podataka, mogućnost prilaganja datoteke s pojašnjenjem ili poveznice na web stranicu s objašnjenjem, ponuđene opcije odgovora, CAPTCHA ili reCAPTCHA, nalaženje trenutnog mjesta pomoću lokacije i strelica za povećanje ili smanjenje brojčane vrijednosti. Obavezna polja su izrazito važna zbog toga što korisnici moraju unijeti odgovor za obavezno pitanje kako bi mogli predati ispunjeni web-obrazac. Na taj se način mogu prikupiti odgovori na sva važna pitanja koja su bitna za određeni web-obrazac te je tako svrha web-obrasca ispunjena. Mogućnost prilaganja datoteke ili poveznice olakšava posao osobi koja analizira rezultate web-obrasca jer se može referirati na dodatno objašnjenje u datoteci ili na poveznici. Ponuđene opcije odgovora pomažu u automatskom sortiranju web-obrazaca; web-obrasci se sortiraju po ponuđenim odgovorima tako da su svi obrasci s istim odgovorom na to pitanje u istoj grupi te se onda obrasci lakše analiziraju. Uz to, korisnik je prisiljen odabrati jedan od ponuđenih odgovora, a ne može unijeti svoj odgovor koji može biti nevezan za to pitanje ili nepotpun. CAPTCHA ili reCAPTCHA služi za provjeru je li korisnik koji popunjava web-obrazac čovjek ili automatizirani robot. Time se preveniraju neželjene poruke i zloupotreba web-obrazaca. Nalaženje mjesta pomoću lokacije pomaže korisniku kada je potrebno unijeti točnu lokaciju tako što se to odvija automatski pa korisnik ne treba unositi naziv svoje točne lokacije. To korisnicima nekada može biti problem ako se nalaze na mjestu koje ne poznaju i zbog toga ne znaju svoju točnu lokaciju. Strelica za povećanje ili smanjenje brojčane vrijednosti može biti korisna kada korisnik treba odabrati količinu nečega ili unijeti broj u određenom rasponu. Strelica olakšava unijeti preciznu vrijednost bez potrebe za tipkovnicom. Ponuđeni kalendar za unos datuma korisnicima olakšava unos podataka tako što se korisniku prikaže kalendar na kojemu može odabrati određeni datum. Na taj način korisnik ne treba otvarati svoj kalendar i provjeriti datum.

Uočeni nedostaci u web-obrascima su sljedeći: neka su polja neobavezna, teško uočljiva pozicija web-obrasca na web stranici, postoji previše pitanja te je strelica za povećanje ili smanjenje brojčane vrijednosti jako mala. Neobavezna polja mogu predstavljati problem

jer dovode do nedostatka informacija za rješavanje upita ili problema. Ako je polje neobavezno, ali je bitan odgovor na to pitanje kako bi prikupili sve potrebne informacije za kasniju analizu, obrada podataka ne može biti kvalitetno napravljena.

Važna je i lokacija web-obrasca; ako je web-obrazac na teško uočljivoj poziciji na web stranici, to također može uzrokovati problem. Korisnik možda neće primijetiti web-obrazac te ga zbog toga neće ni ispuniti. To će možda korisniku uzrokovati negativne emocije oko proizvoda te će ga prestati koristiti. Nadalje, ako web-obrazac sadrži puno pitanja, postoji mogućnost da korisnik neće ispuniti web-obrazac zbog količine vremena koje mu je potrebno za odgovaranje na svako polje. Na taj način organizacija neće prikupiti mišljenja i probleme korisnika te neće moći napredovati. Također, ako je strelica za povećanje ili smanjenje brojčane vrijednosti jako mala, moguće je da ju korisnik neće primijetiti i iskoristiti.

Iz navedenog zaključeno je da je izrazito važno imati obavezna polja koja su nužna za prikupljanje svih potrebnih informacija za obradu web-obrasca. Također je korisno kada neka pitanja sadrže već ponuđene odgovore kako bi se kasnije web-obrasci mogli lakše sortirati i obraditi. Bitno je i osigurati da korisnik ne može namjerno unijeti netočan odgovor koji bi otežao obradu. Uz to je korisno omogućiti korisniku da priloži datoteku ili poveznicu kako bi lakše pojasnio svoje odgovore ili pitanja.

2. Potrebe korisnika

Web-obrasci za prijavu korisničkih neispravnosti omogućuju korisnicima da jednostavno i brzo prijave grešku koju su pronašli u aplikaciji. Pri dizajniranju web-obrasca važno je razmotriti potrebe korisnika kako bi se korisnicima omogućilo što bolje iskustvo.

Primarni korisnici ICT-AAC aplikacija su djeca s teškoćama u razvoju kojima je jako bitno da neprekidno mogu koristiti aplikacije kako bi kontinuirano učili i igrali se. Zato je izrazito važno da roditelji, pedagozi, logopedi, psiholozi i sve osobe koje pomažu i rade s djecom mogu brzo i jednostavno prijaviti grešku na koju su naišli. Na taj se način problem može brže riješiti kako bi djeca mogla nastaviti s učenjem.

Jednostavni i intuitivni web-obrasci poboljšavaju korisničko iskustvo jer korisnici lako razumiju kako ispuniti obrazac bez potrebe za dodatnim objašnjenjima. Na taj se način smanjuje mogućnost da se korisnik osjeća frustrirano zbog nejasnog web-obrasca. Važno je da se u web-obrascu traže samo nužne informacije zbog efikasnijeg ispunjavanja jer se time štedi vrijeme korisnika pa su i korisnici zadovoljniji. Ako obrazac sadrži previše pitanja, postoji mogućnost da korisnik odustane od ispunjavanja jer to zahtijeva previše truda i vremena ili da ne odgovori kvalitetno na pitanja zbog žurbe. Važno je da je web-obrazac pregledan i jednostavan kako bi se smanjila vjerojatnost rađanja greški pri unosu podataka. Kada su pitanja i polja za unos podataka jasno označena te postoje upute ili primjer ispunjavanja, manja je vjerojatnost da korisnik napravi grešku pri unosu. Prilaganje datoteka korisnicima pomaže u objašnjenju problema jer je nekad korisnicima jednostavnije slikom ili videom prikazati problem nego ga objasniti riječima. Bitno je osigurati poruku o potvrđivanju primitka web-obrasca kako bi korisnik bio siguran da je njegov ispunjeni web-obrazac poslan. Također je bitno da se privatni podaci korisnika sigurno čuvaju.

Česta ažuriranja operacijskih sustava i pojave novih prijenosnih uređaja utječu na učestalost grešaka u web-aplikacijama, igricama, web stranicama itd. Dolazi do razvoja kompleksnijih tehnoloških rješenja koja mogu biti izazovna za razumijevanje i korištenje te se zbog toga povećava broj potencijalnih grešaka. Također, na današnjem tržištu postoji širok raspon uređaja, platformi i operacijskih sustava te je potrebno razviti proizvod koji podržava što veći broj uređaja. To može biti složen proces te su greške moguće.

Nadalje, zbog specifičnosti svakog uređaja i operacijskog sustava potrebno je znati podatke o uređaju na kojem se dogodila greška. Zato je važno da web-obrazac sadrži obavezna pitanja o vrsti uređaja, verziji operacijskog sustava i modelu uređaja kako bi korisnici unijeli potrebne podatke nakon čega članovi organizacije čiji je web-obrazac mogu ispraviti pronađenu grešku.

3. Opis razvijene web-aplikacije za praćenje korisničkih prijava neispravnosti

Zbog brzog razvoja tehnologija i čestih ažuriranja operacijskih sustava greške u aplikacijama su česte i neizbježne. Važno je korisnicima omogućiti jednostavan i brz način prijave pronađenih grešaka kako bi greške bile ispravljene u što kraćem vremenskom roku i time se omogućilo daljnje korištenje aplikacije bez poteškoća. Web-aplikacija je razvijena kako bi se njome olakšala prijava korisničkih neispravnosti u ICT-AAC aplikacijama te omogućio pregled poslanih prijava kako bi se neispravnosti mogle riješiti.

Cilj je web-aplikacije pružiti korisnicima jednostavan i intuitivan način prijave neispravnosti koje su pronašli kako bi članovi kompetencijske mreže ICT-AAC iz poslanih prijave mogli efikasno riješiti problem koji se dogodio. Na taj način djeca nemaju veliki vremenski prekid u kojem nisu u mogućnosti učiti te im je omogućen konzistentan rad i razvoj.

Budući da su primarni korisnici ICT-AAC aplikacija djeca s teškoćama u razvoju, web-aplikacija je primarno namijenjena roditeljima, pedagogima, logopedima i drugim osobama koje djeci pomažu koristiti aplikacije. Stoga korisnici obrasca mogu biti starije osobe koje nemaju puno iskustva u korištenju različitih tehnologija te je zbog toga važno da je web-obrazac intuitivan i lagan za korištenje za sve osobe neovisno o životnoj dobi i iskustvu uporabe različitih tehnologija. Primjerice, zato ispod nekih pitanja postoji primjer odgovora kako bi korisnicima bilo jasno koji odgovor se traži.

Korisničko sučelje treba biti ugodno, intuitivno i učinkovito za korištenje. Zbog toga su elementi pažljivo raspoređeni te je pomno odabrana paleta boja i veličina fonta. Raspored elemenata utječe na upotrebljivost i iskustvo, stoga je bilo važno imati logičan raspored kako bi korisnici brzo pronašli sve informacije i ispunili web-obrazac. Primjerice, važno je bilo grupirati slična pitanja kako bi korisnicima olakšali unos pitanja i poboljšali iskustvo. Boje imaju veliku ulogu u vizualnoj privlačnosti web stranice. Različite boje različito utječu na osjećaje i iskustvo korisnika. Zato su pažljivo odabrane odgovarajuće boje za web-aplikaciju. Veličina fonta utječe na čitljivost i pristupačnost te kontrast između teksta i pozadine. Odabrana je odgovarajuća veličina fonta kako bi korisnici mogli lako pročitati

sav sadržaj. Također je bilo ključno osigurati da je web-aplikacija responzivna kako bi ju korisnici mogli koristiti na različitim uređajima bez ikakvih poteškoća.

Web-obrazac sadrži obavezna i neobavezna pitanja: obavezna su pitanja ona koja su nužna za prikupljanje svih ključnih informacija, a neobavezna pitanja samo dodatno objašnjavaju situaciju. Iako web-obrazac ne bi trebao biti predugačak kako korisničko iskustvo ne bi bilo negativno, ipak je važno imati obavezna pitanja jer se pomoću njih mogu prikupiti sve potrebne informacije. Isto je tako dobro imati i nekoliko neobaveznih pitanja kako bi korisnik imao mogućnost odgovoriti samo ako želi, ali time i unijeti dodatne informacije. Na neka pitanja ponuđeni su odgovori kako bi se ograničilo korisnika na odabir jednog od odgovora i time mu se onemogućilo da unese svoj odgovor, koji može biti netočan. U web-obrascu korisniku je omogućeno prilaganje datoteka kao bi lakše objasnio svoj problem.

Bilo je potrebno osigurati i spremanje podataka iz web-obrazaca kako bi se ti podaci mogli prikazivati u drugom dijelu aplikacije. To je ostvareno pomoću XML datoteka.

Drugi dio web-aplikacije služi za prikaz poslanih web-obrazaca te je namijenjen članovima kompetencijske mreže ICT-AAC koji su zaduženi za ispravljanje grešaka koje su se dogodile. Poslani web-obrasci organizirani su na stranici za prikaz kako bi se olakšalo snalaženje osobama koje se bave ispravljanjem grešaka. Također je odabran font odgovarajuće veličine i boje kako bi se stvorio kontrast između teksta i pozadine i tako olakšala čitljivost.

4. Tehničke značajke razvijene web-aplikacije

Web-aplikacija je razvijena pomoću tehnologije *React* za razvoj korisničkog sučelja te XML datoteke za pohranu i prikaz podataka. Odabrana je XML datoteka za pohranu podataka zbog jednostavnosti spremanja i čitanja podatka jer za razliku od baze podataka ne zahtjeva instalaciju i konfiguraciju. Web-aplikacija ne zahtijeva složene funkcionalnosti kao što su transakcije, upiti i pohrana velikih količina podataka te je zbog toga odabrana XML datoteka. U datoteku se spremaju podaci web-obrasca te se čitaju kako bi se prikazali. Za lokalni poslužitelj koji čita i zapisuje podatke u XML datoteku koristi se tehnologija *Express.js*.

Razvoj web-aplikacije podijeljen je u nekoliko aktivnosti; prvo je stvorena stranica za web-obrazac na hrvatskome jeziku te je instaliran i pokrenut lokalni poslužitelj na kojem se nalazi XML datoteka radi lakšeg razvoja. Omogućen je upis podatak u datoteku. Nakon toga je stvorena stranica za prikaz podataka iz XML datoteke te omogućen prikaz. Zatim je dodana opcija odabira između hrvatskog i engleskog jezika te je stvoren web-obrazac na engleskom jeziku. Na kraju je dodan dio za informacije na hrvatskom i engleskom jeziku. Na kraju razvoja, testirana je aplikacija te su ispravljene pronađene greške.

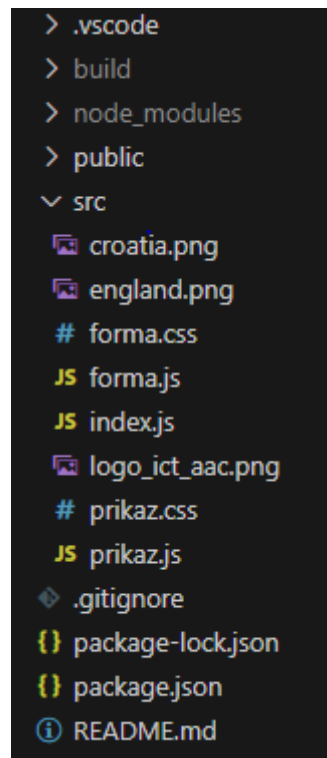
4.1. Tehnologije i alati

React je *JavaScript* biblioteka koja se koristi za izgradnju korisničkog sučelja za web-aplikacije. Kreirao ju je Facebook 2013. godine [16]. Omogućuje stvaranje ponovo upotrebljivih komponenata sučelja što olakšava održavanje aplikacija. *React DOM* je dodatan paket koji omogućava renderiranje komponenata u stvarni DOM te sinkronizaciju između virtualnog i stvarnog DOM-a. Podržava *JSX* sintaksu koja omogućuje kombiniranje *HTML* i *JavaScripta* u jednom kodu. Ima veliku i aktivnu zajednicu korisnika koja olakšava i poboljšava razvoj aplikacija. Budući da je jednostavan i fleksibilan za korištenje, jedan je od najpoznatijih alata za razvoj korisničkog sučelja za aplikacije. *Express.js* je minimalistički i fleksibilan web okvir za *Node.js* koji omogućuje jednostavno i brzo kreiranje web i mobilnih aplikacija. Pruža sustav usmjeravanja koji omogućuje definiranje URL ruta i njihovo povezivanje s različitim *HTTP* metodama.

4.2. Arhitektura razvijene web-aplikacije

Korisničko sučelje je razvijeno pomoću *Reacta*. Komponente su organizirane prema hijerarhiji koja održava organizaciju i funkcionalnost aplikacije što omogućava lakše održavanje i nadogradnju aplikacije. Organizacija komponenata prikazana je grafički (Slika 4.1 Hijerarhija web-aplikacije). Glavna komponenta *index.js* upravlja rutama korištenjem *React Router*a. Korišten je *BrowserRouter* kako bi se definirale rute pomoću *Routes* komponente `<Route path="/" element={<Forma />} />`. Svaka ruta definira putanju i određenu komponentu koja će se prikazati za tu rutu. *Forma.js* i *prikaz.js* predstavljaju pojedinačne stranice web-aplikacije. Unutar *indeks.js-a* koristi se *path* za definiranje navigacijskih puteva. Komponenta *forma.js* predstavlja glavnu stranicu na kojoj se nalazi web-obrazac za prijavu korisničkih neispravnosti. Komponenta *prikaz.js* služi za prikaz prijavljenih neispravnosti kako bi sve prijave bile na jednom mjestu i prikazane pregledno te se time omogućilo što brže ispravljanje neispravnosti.

Pokretanjem web-aplikacije pokreće se *forma.js* gdje se nalazi web-obrazac. Prelazak između *forma.js* i *prikaz.js* nije omogućen zašto što je *prikaz.js* namijenjen samo za osobe čiji je posao ispraviti greške koje su se dogodile.



Slika 4.1 Hijerarhija web-aplikacije

Stanje u *React* komponentama predstavlja dinamičke podatke koji se mogu mijenjati tijekom vremena i utječu na prikaz te komponente. Omogućuje da su komponente interaktivne i dinamičke stoga reagiraju na korisnikov unos i druge promjene. U komponenti *forma.js* koristi se stanje kako bi se pratile vrijednosti za polja za unos teksta ili odabir ponuđenih opcija. Kada korisnik unese ili odabere neku vrijednost, stanje se automatski ažurira. Za polja za unos teksta koristi se stanje kako bi se pratila trenutna vrijednost unesenog teksta. Kada korisnik počne unositi slova, stanje se ažurira na trenutnu unesenu vrijednost. Npr. u polju za unos pitanja: „Unesite vrstu internetskog preglednika”, koristi se stanje kako bi se pratila vrijednost tog polja i ažuriralo stanje svaki put kada korisnik unese neko slovo ili izbriše neko slovo. Za polja za odabir opcija poput padajućeg izbornika ili radio gumba također se koristi stanje kako bi se pratila trenutno odabrana opcija. Kada korisnik promijeni opciju, stanje se ažurira. To je prikazano u sljedećem primjeru (Kôd 4.1):

```
const [browserType, setBrowserType] = useState("");

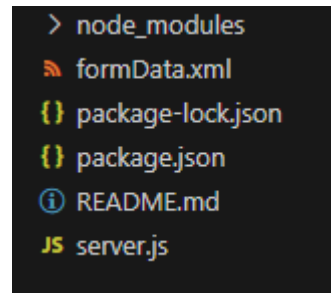
const handleBrowserTypeChange = (event) => {
  setBrowserType(event.target.value);
};

<label>Unesite vrstu internetskog preglednika: <span
className="required">*</span></label>
<input
  type="text"
  value={browserType}
  onChange={handleBrowserTypeChange}
  required
  onInvalid={(e) => e.target.setCustomValidity('Molimo
unesite vrstu internetskog preglednika.')}
  onInput={(e) => e.target.setCustomValidity('')}
/>
```

Kôd 4.2 – Primjer koda u kojem se koristi stanje kako bi se pratila trenutna vrijednost unesenog teksta

Lokalni poslužitelj je instaliran i pokrenut pomoću *express.js* kako bi se na poslužitelju mogla nalaziti XML datoteka u koju se upisuju i iz koje se čitaju podatci. Hijerarhija na lokalnom poslužitelju prikazana je grafički (Slika 4.2 Hijerarhija lokalnog poslužitelja). Web-aplikacija upisuje i čita podatke iz XML datoteke. Ovakva komunikacija vrši se

putem HTTP zahtjeva na poslužitelj koji sadrži XML datoteku. Kada se spremaju podatci u XML datoteku, podatci se pretvaraju u XML format te se šalju HTTP POST zahtjevom na poslužitelj gdje se nalazi datoteka. Poslužitelj zatim pohranjuje te podatke u XML datoteku. Kada se žele pročitati podatci, onda se HTTP GET zahtjev šalje na odgovarajuću putanju na poslužitelj. Poslužitelj odgovara XML datotekom koja sadrži tražene podatke. Web-aplikacija zatim pretvara dobivene podatke u odgovarajući oblik koji je potreban za obradu.



Slika 4.2 Hijerarhija lokalnog poslužitelja

Vizualni dio web-aplikacija ostvaren je pomoću CSS-a. Važno je bilo odabrati boje koje su ugodne i međusobno se dobro upotpunjuju. Odabrane su dvije nijanse plave boje za web-aplikaciju zato što je prema istraživanjima [17] plava smirujuća i spokojna boja koja pruža osjećaj povjerenja, sigurnosti i pouzdanosti. Plava je boja odabrana i zato što potiče produktivnost, a cilj je bio postići da korisnici prijavu ispune brzo i efikasno. Također je bilo važno odabrati vrstu i veličinu fonta za sve dijelove web-aplikacije jer utječu na korisnikov doživljaj prilikom interakcije sa samom web-aplikacijom zbog čitljivosti i vizualnog dojma. Elementi su pažljivo raspoređeni uzimajući u obzir kriterije jasnoće i estetike.

4.3. Funkcionalnosti razvijene web-aplikacije

Funkcionalni zahtjevi web-aplikacija za praćenje korisničkih prijava neispravnosti su:

- Unos podataka
- Prilaganje datoteka
- Validacija unosa
- Poruka o grešci ako obavezno polje nije ispunjeno
- Prijenos podataka

- Prikaz podataka

Neke podatke korisnik mora unijeti, a neke može samo ako želi. Korisnik mora unijeti ime aplikacije u kojoj je uočio grešku, odabrati dio aplikacije u kojem je uočio grešku, detaljno opisati prijavljenu grešku, odabrati vrstu uređaja, unijeti model uređaja i unijeti vrstu operacijskog sustava. Može priložiti datoteku i unijeti svoju e-adresu. Korisnik može učitati sliku ili video greške. Može učitati maksimalno pet datoteka te su dopuštene samo datoteke s ekstenzijama *jpg*, *jpeg*, *png*, *bmp*, *mp4*. Odabrane su ove ekstenzije jer se najčešće koriste i podržava ih većina preglednika. Također, korisnik mora moći poslati prijavu greške pritiskom na gumb „Pošalji” ili „Submit”.

Korisnik mora popuniti sva obavezna polja (ime aplikacije, dio aplikacije u kojem je greška uočena, opis greške, vrstu uređaja, model uređaja, vrstu operacijskog sustava) kako bi mogao poslati web-obrazac. Ako korisnik odabere opciju „Ostalo” za ime aplikacije ili dio aplikacije u kojem je uočena greška, prikazuje se novo obavezno pitanje s poljem za unos odgovora i korisnik mora unijeti odgovor. Ako korisnik ne ispuni sva obavezna polja, treba mu se prikazati poruka o grešci.

Pritiskom na gumb „Pošalji” ili „Submit” podaci se trebaju uspješno poslati i upisati u XML datoteku. Na stranici za prikaz podataka podaci se trebaju uspješno pročitati iz XML datoteke te prikazati. Pitanja na engleskom jeziku ista su kao i pitanja na hrvatskome jeziku.

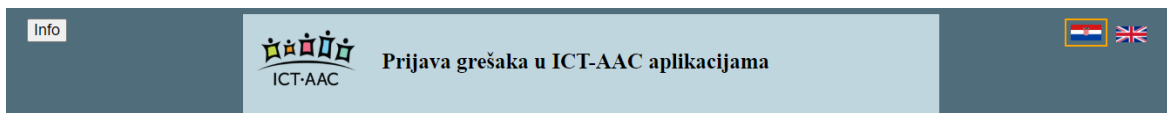
4.4. Elementi razvijene web-aplikacije

Elementi web-aplikacije su stranica za web-obrazac i stranica za pregled rezultata web-obrazac. Na stranici za web-obrazac nalazi se sami web-obrazac, gumb za info dio i gumbi za odabir jezika (hrvatski ili engleski jezik), koji su prikazani na sljedećoj slici (Slika 4.3).



Slika 4.3 Gumbi za odabir jezika

Na vrhu stranice raspored je sljedeći: lijevo se nalazi info gumb, desno gumbi za odabir jezika te u sredini logo i naslov (Slika 4.4).



Slika 4.4 Prikaz gornjeg dijela stranice za web-obrazac

Kada se pritisne info gumb, prikazuje se nekoliko rečenica s informacijama o web-aplikaciji na hrvatskom ili engleskom jeziku (Slika 4.5), ovisno o tome koji je jezik odabran. Informacije se zatvaraju pritiskom na gumb „Zatvori” ili „Close”. Pritiskom na ikonu hrvatske ili engleske zastave stranica se postavlja na odabrani jezik.



Slika 4.5 Informacije na hrvatskom jeziku

Web-obrazac se sastoji od različitih pitanja, a odgovori na njih su važni za ispravak određene greške. Postoje pitanja na koje treba upisati odgovor i ona gdje treba izabrati jedan od ponuđenih odgovora (Slika 4.6).



Slika 4.6 Odabir jednog od ponuđenih odgovora

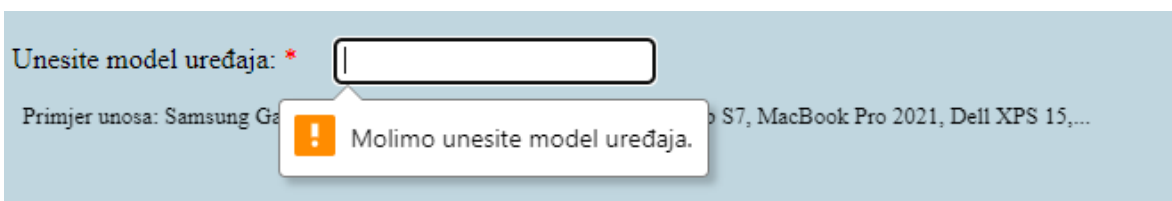
Također postoji mogućnost prilaganja slike ili videa u web-obrazac kako bi se lakše razumjelo gdje je greška nastala i kako izgleda (Slika 4.7).

Ako ste u mogućnosti, napravite presliku ekrana s prikazanom greškom ili slikajte grešku ili snimite video, te uploadajte datoteku

(maksimalno 5 datoteka):

Slika 4.7 Prilaganje datoteke

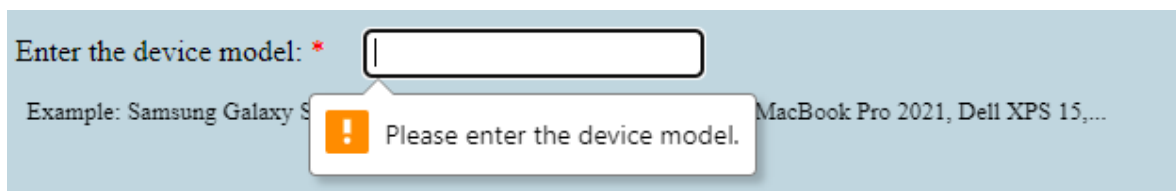
Isto tako, neka su pitanja obavezna te se na njih mora odgovoriti kako bi se web-obrazac predao, a neka su neobavezna i nije potrebno odgovoriti na njih da se preda web-obrazac. Ako korisnik ne odgovori na sva obavezna pitanja pri pokušaju slanja web-obrasca, prikazat će se poruka ispod pitanja na koje nije odgovorio kako bi se korisnika obavijestilo da treba odgovoriti na sva obavezna pitanja prije slanja web-obrasca (Slika 4.8 i Slika 4.9).



Unesite model uređaja: *

Primjer unosa: Samsung Galaxy S7, MacBook Pro 2021, Dell XPS 15,...

Slika 4.8 Primjer poruke za obavezno polje na hrvatskom jeziku

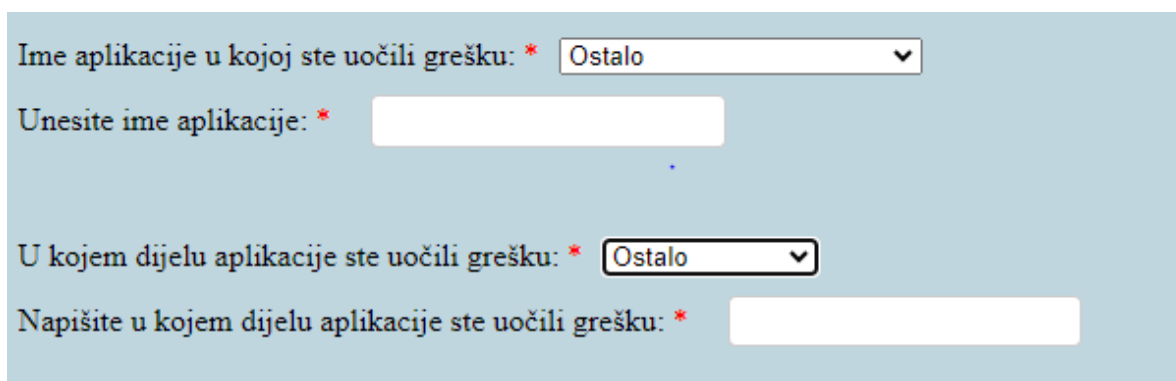


Enter the device model: *

Example: Samsung Galaxy S7, MacBook Pro 2021, Dell XPS 15,...

Slika 4.9 Primjer poruke za obavezno polje na engleskom jeziku

Ako se na pitanje „Ime aplikacije u kojoj ste uočili grešku:” ili „U kojem dijelu aplikacije ste uočili grešku:” odgovori „Ostalo”, prikazuje se novo obavezno pitanje s poljem za unos (Slika 4.10). Isto tako, ako se na pitanje „Odaberite vrstu uređaja:” odgovori „Web aplikacija” prikazuje se novo obavezno pitanje s poljem za unos (Slika 4.11).



Ime aplikacije u kojoj ste uočili grešku: *

Unesite ime aplikacije: *

U kojem dijelu aplikacije ste uočili grešku: *

Napišite u kojem dijelu aplikacije ste uočili grešku: *

Slika 4.10 Primjer prikaza novih pitanja ovisno o odabranom odgovoru

Select the type of device: *

Android
 iOS
 Web application



Enter the type of web browser: *

Example: Google Chrome, Mozilla Firefox, Safari,...

Slika 4.11 Primjer prikaza novog pitanja ovisno o odabranom odgovoru na engleskom jeziku

Na stranici za prikaz (Slika 4.12) nalazi se tablica. U stupcima su navedena pitanja iz web-obrasca, a svaki redak predstavlja korisničke podatke iz jednog unosa. Pritiskom na gumb „Prikaži u novom prozoru” otvara se novi prozor u pregledniku na kojem se nalazi ta datoteka.

Podaci iz XML datoteke:

Ime aplikacije	Ostalo ime aplikacije	Mjesto aplikacije	Ostalo mjesto aplikacije	Opis problema	Mediji	Vrsta uređaja	Vrsta internetskog preglednika	Model uređaja	Verzija operacijskog sustava	E-adresa
Ostalo	Domino+	Ostalo	Izlaz iz igre	kada stisnem ...	 Prikaži u novom prozoru	Web	Google chrome	Samsung galaxy a32 5g	Android 11	domino@fer.hr
Other	Domino+	Other	Main scene	when I	 Prikaži u novom prozoru	Web	Firefox	Sony experia	Android 10	yes@fer.hr

Slika 4.12 Stranica za prikaz podataka iz XML datoteke

4.5. Implementacija razvijene web-aplikacije

Kod je napisan u *Reactu*, *indeks.js* služi za upravljanje rutama, *forma.js* i *forma.css* su zaduženi za stranicu gdje se nalazi web-obrazac, a *prikaz.js* i *prikaz.css* su zaduženi za prikaz podataka XML datoteke.

Prikaz određenog jezika napravljen je na sljedeći način: ako je odabran hrvatski jezik, prikazuje se hrvatski dio, a ako je odabran engleski jezik, prikazuje se engleski dio:

language === 'hr' && (...)}{language === 'en' && (...)}. Početna vrijednost varijable language je hr, a ako se klikne na englesku zastavu zove se funkcija handleLanguageChange koja postavlja vrijednost varijable language na eng. Isto vrijedi za hrvatsku zastavu. Funkcija handleLanguageChange postavlja vrijednosti svih varijabla na početnu vrijednost kako bi nakon promjene jezika web-obrazac bio u početnom stanju.

```
const handleLanguageChange = (selectedLanguage) => {
  setAppName("");
  setCustomAppName("");
  setLocation("");
  setCustomLocation("");
  setDescription("");
  setMedia([]);
  setDeviceType("");
  setBrowserType("");
  setDeviceModel("");
  setOsType("");
  setEmail("");
  setHelper("");
  setLanguage(selectedLanguage);
};
```

Kôd 4.3 - Funkcija handleLanguageChange

Na sličan je način napravljen prikaz info dijela i web-obrasca. Kada se otvori stranica, prikazan je web-obrazac. Stanja varijabli su setShowForm(true) i setShowPopup(false) te kada se klikne na „info” gumb zove se funkcija hadleInfoClick koja postavlja vrijednost setShowForm(false) i setShowPopup(true). Klikom na gumb „zatvori” ili „close” poziva se funkcija handleClosePopup koja postavlja varijable na početne vrijednosti setShowForm(true) i setShowPopup(false).

Obavezne komponente postignute su tako da je u komponentu koja treba biti obavezna postavljen required te je pomoću onInvalid={ (e) => e.target.setCustomValidity('')} postavljena prilagođena poruka, a pomoću onInput={ (e) => e.target.setCustomValidity('')} uklonjena prilagođena poruka nakon što je komponenta ispunjena.

```

<label>Unesite verziju operacijskog sustava: <span
className="required">*</span></label>
<input
type="text"
value={osType}
onChange={handleOsTypeChange}
required
onInvalid={(e) => e.target.setCustomValidity('Molimo
unesite verziju operacijskog sustava.')}
onInput={(e) => e.target.setCustomValidity('')}
/>

```

Kôd 4.4 – Primjer obaveznog pitanja sa prilagođenom porukom

Validacija e-adrese postignuta je funkcijom koja provjerava sadrži li e-adresa sve potrebne dijelove kako bi bila ispravnog oblika. Ovaj regularni izraz postavlja određene uvjete koje e-adresa mora zadovoljiti kako bi bila ispravna. Lokalni dio e-adrese mora sadržavati berem jedno od dopuštenih znakova (slovo, broj, točku, donju crticu, postotak, plus ili minus). Iza lokalnog dijela mora se nalaziti znak @, nakon znaka @ nalazi se ime lokalne domene. Ime lokalne domene treba sadržavati barem jedan znak (slovo, broj, točku ili crticu), zatim točku, zatim najmanje dva slova.

```

const isValidEmail = (email) => {
const emailRegex = /^[a-z0-9._%+-]+@[a-z0-9.-]+\.[a-z]{2,}$/i;
return emailRegex.test(email);
};

```

Kôd 4.5 – Funkcija za validaciju e-adrese

Korisnik može priložiti presliku ekrana, sliku ili video greške, ali je ograničen na maksimalno pet datoteka, video veličine do 20MB, a ekstenzije slikovnih i video datoteka su također ograničene na *jpg*, *jpeg*, *png*, *bmp*, *mp4*. Ekstenzije su provjerene tako da ako `if(!allowedExtensions.includes(extension))` ekstenzija priložene datoteke nije jedna od ponuđenih ekstenzija, prikaže se poruka „Nedopuštena datoteka: {fileName}. Molimo odaberite datoteke s ekstenzijama: jpg, jpeg, png, bmp, mp4.” ili „Invalid file: {fileName}. Please choose files with extensions: jpg, jpeg, png, bmp, mp4.”, ovisno o odabranom jeziku, a priložena se datoteka ne spremi. Ograničenje videa postignuto je tako da se provjerava `if (file.type.includes('video') && file.size > MAX_VIDEO_SIZE_MB * 1024 * 1024)`. Ako je priloženi video veći od dopuštene veličine, prikaže se poruka „Veličina videa {fileName} prelazi

maksimalno dopuštenih {maxVideoSize} MB.” ili „The size of the video {fileName} exceeds the maximum allowed {maxVideoSize} MB.” ovisno o odabranom jeziku i video se ne sprema. Provjera je li ukupan broj datoteka veći od pet postignuta je tako da se provjerava bi li dodavanje nove datoteke premašilo maksimalno dopušteni broj datoteka `if (filesArray.length + media.length > 5`. Ako je dodavanje nove datoteke nemoguće zbog broja već dodanih datoteka, prikazuje se poruka „Možete odabrati maksimalno 5 datoteka.” ili „You can select a maximum of 5 files.”, ovisno od odabranom jeziku te se priložena datoteka ne sprema.

Prikaz slike sa gumbom „Prikaži u novom prozoru” na stranici za prikaz napravljen je tako da je `` prosljeđen `src` i `alt` koji je dobiven čitanjem XML datoteke. Ispod slike se nalazi gumb. Kada se na njega klikne, poziva se funkcija `debugImageBase64` koja prima jedan argument *base64URL* koji je *string* i predstavlja URL slike kodirane u *base64* formatu. Otvara novi prozor `var win = window.open()`; u koji upiše HTML sadržaj tj. `` element.

```
function debugImageBase64(base64URL) {
    var win = window.open();
    win.document.write('');
}
```

Kôd 4.6 – Funkcija `debugImageBase64`

Prikaz videa s gumbom „Prikaži u novom prozoru” napravljen je na sličan način; `<video>` koji unutar sebe ima element `<source>` prosljeđen je `src` koji je dobiven čitanjem XML datoteke. Pritiskom na gumb poziva se funkcija `debugVideoBase64` koja prima jedan argument *base64URL* koji je *string* i predstavlja URL videa kodiranog u *base64* formatu. Otvara novi prozor u koji upiše sadržaj `<source>` elementa.

```
function debugVideoBase64(base64URL) {
    var win = window.open();
    win.document.write('<video controls style="width:100%;
    height:auto;"><source src="' + base64URL + '"
    type="video/mp4"></video>');
}
```

Kôd 4.7 – Funkcija `debugVideoBase64`

5. Testiranje razvijene web-aplikacije

Testiranje web-aplikacije važan je dio razvoja jer osigurava da web-aplikacija radi na različitim platformama i u različitim scenarijima. Tijekom razvoja ove web-aplikacije provedeno je testiranje na različitim web preglednicima i u različitim scenarijima. Aplikacija je testirana na preglednicima Google Chrome, Mozilla Firefox i Microsoft Edge. Svaki preglednik ima svoje specifičnosti pa postoji mogućnost da se web-aplikacija ponaša drugačije na svakom od njih. Zbog toga su testirane sve funkcionalnosti na svim preglednicima. Uočeno je da jezik poruke ispod obaveznog pitanja ovisi o postavljenom jeziku preglednika. Zbog toga je svaka poruka prilagođena tako da bude na jeziku koji je odabran na web-obrascu, tj. tako da su poruke neovisne o jeziku preglednika. Također su testirani različiti scenariji kako bi se osiguralo da web-aplikacija ispravno radi za različite situacije. Testirano je kako se web-aplikacija ponaša ako se ne ispune sva obavezna polja i javlja li odgovarajuće poruke za svako obavezno polje. To je provedeno tako da se pokušao poslati web-obrazac kada nije odgovoreno na nijedno pitanje, kada je odgovoreno na samo jedno pitanje, itd. i na taj način su isprobane različite kombinacije. Testirana je responzivnost kako bi se provjerilo kako se web-aplikacija prilagođava različitim veličinama uređaja. Promijenjen je raspored elemenata ovisno o veličini uređaja. Prilaganjem različitih datoteka različitih ekstenzija i veličina, testirano je spremanje i prikaz podataka te prikaz datoteke u novom prozoru. Uočen je problem u prikazu datoteka u novom prozoru zbog ograničenja veličine te je greška ispravljena uklanjanjem tog ograničenja. Također su provjerene sve poruke i sav tekst na web-aplikaciji kako bi bili gramatički i pravopisno ispravni. Navedenim temeljnim testiranjem pronađeni su i ispravljani različiti problemi kako bi se omogućilo kvalitetno korisničko iskustvo.

6. Primjer korištenja web-aplikacije

Kada korisnik otvori stranicu za web-obrazac, prikaže mu se web-obrazac na hrvatskom jeziku. Ukoliko želi web-obrazac na engleskom jeziku, mora odabrati englesku zastavu u gornjem desnom kutu. Nakon što je korisnik odabrao određeni jezik, može krenuti s odgovaranjem na pitanja (Slika 6.1). Potrebno je odgovoriti na sljedeća pitanja kako bi se prikupile potrebne informacije za rješavanje greške. Ako je odabran hrvatski jezik, prvo pitanje je „Ime aplikacije u kojoj ste uočili grešku:”. Korisniku su ponuđeni odgovori i može odabrati jednu od ponuđenih aplikacija. U slučaju da u ponuđenim aplikacijama nema onu koju treba, može odgovoriti „Ostalo”. Ako je odgovorio „Ostalo”, prikazuje se novo pitanje „Unesite ime aplikacije:”, gdje korisnik treba upisati ime aplikacije u kojoj je uočio grešku. Sljedeće je pitanje „U kojem dijelu aplikacije ste uočili grešku:”. Na ovo pitanje ima ponuđene odgovore te, ako ne pronalazi svoj odgovor u ponuđenima, može odgovoriti „Ostalo”. Ako je odgovorio „Ostalo”, prikazuje se novo pitanje „Napišite u kojem dijelu aplikacije ste uočili grešku:” na koje treba odgovoriti. U sljedećem pitanju „Detaljno opišite prijavljenu grešku:” treba opisati grešku koja se dogodila. Korisnik može priložiti presliku ekrana s prikazom greške ili sliku greške ili snimljeni video.

Nakon toga treba odabrati vrstu uređaja koji je koristio kada je uočio grešku. Na pitanje „Odaberite vrstu uređaja:” može odabrati jednu od tri opcije: *Android*, *iOS* ili *Web aplikacija*. Ako odabere *Web aplikacija*, prikazuje se novo pitanje „Unesite vrstu internetskog preglednika:” na koje mora upisati odgovor. Zatim na pitanja „Unesite model uređaja:” i „Unesite verziju operacijskog sustava:” treba upisati odgovor. Korisniku se nudi mogućnost da unese svoju e-adresu kako bi ga se moglo kontaktirati u slučaju da su potrebne dodatne informacije oko prijavljene greške.

Ako korisnik ne odgovori na sva obavezna pitanja, prikazuje se poruka ispod pitanja na koje nije odgovorio (Slika 6.2). Nakon što korisnik odgovori na sva obavezna pitanja, može stisnuti gumb „Pošalji”. Ako je web-obrazac uspješno poslan, prikazuje se poruka o uspjehu, a, ako je došlo do greške, prikazuje se poruka o grešci. U slučaju da je odabran engleski jezik, sva su pitanja ista samo su na engleskom jeziku. Kada je web-obrazac uspješno poslan, podaci se spremaju u XML datoteku.

Info

ICT-AAC

Prijava grešaka u ICT-AAC aplikacijama

Ime aplikacije u kojoj ste uočili grešku: * Domino brojilica

U kojem dijelu aplikacije ste uočili grešku: * Ostalo

Napišite u kojem dijelu aplikacije ste uočili grešku: * Početna scena

Detaljno opišite prijavljenu grešku: * Ne vidi se gumb za igranje igre

Ako ste u mogućnosti, napravite presliku ekrana s prikazanom greškom ili slikajte grešku ili snimite video, te uploadajte datoteku
(maksimalno 5 datoteka): [Prikaži datoteku](#)

Podaci o uređaju

Unesite podatke o uređaju na kojem ste uočili grešku koju prijavljujete. Bez ovih podataka ponekad nismo u mogućnosti ispraviti greške.

Slika 6.1 Primjer korištenja web-obrasca

Unesite podatke o uređaju na kojem ste uočili grešku koju prijavljujete. Bez ovih podataka ponekad nismo u mogućnosti ispraviti greške.

Odaberite vrstu uređaja: *

Android

iOS

Web aplikacija

Unesite model uređaja: * Samsung galaxy a32

Primjer unosa: Samsung Galaxy S21, iPhone 13 Pro, Samsung Galaxy Tab S7, MacBook Pro 2021, Dell XPS 15,...

Unesite verziju operacijskog sustava: *

Ovu informaciju pronađite na sljedećim linkovima: [Kako pronaći verziju operacijskog sustava na uređaju -> Verzija sustava Android.](#)
Za iOS uređaje postupak je sljedeći: [Kako pronaći verziju operacijskog sustava na iOS uređaju.](#)
Za računala ili laptope postupak je sljedeći: [Kako pronaći verziju operacijskog sustava na računaru.](#)

Molimo unesite verziju operacijskog sustava.

Primjer unosa: Android 11, iOS 15, Windows 10, Ubuntu 20.04 LTS,...

Unesite svoju e-adresu kako bismo vas mogli kontaktirati ako trebamo dodatne informacije:

Pošalji

Slika 6.2 Primjer predaje web-obrasca prije nego su sva pitanja ispunjena

Osobe zadužene za ispravljanje pronađenih grešaka mogu otići na drugi dio web-aplikacije gdje su prikazane prijavljene pogreške kako bi saznale sve potrebne podatke za rješavanje greške.

7. Potrebe održavanja

Kako bi se održala dugoročna funkcionalnost, izrazito je važno održavati web-aplikaciju. Održavanje omogućuje otkrivanje i ispravljanje grešaka, prilagodbu s obzirom na nova tehnološka znanja i potrebe korisnika, koje se mogu promijeniti s vremenom. Kako bi web-aplikacija mogla raditi bez velikih prekida, ključno je ispraviti greške ako do njih dođe. Važno je prilagoditi web-aplikaciju s obzirom na promjene u okruženju kao što je nadogradnja operacijskih sustava ili preglednika tako da web-aplikacija ostane funkcionalna i nakon promjena. S vremenom će se razviti nove ICT-AAC aplikacije koje sada nisu ponuđene u prvom pitanju „Ime aplikacije u kojoj ste uočili grešku?”. Zbog toga je jedan od ponuđenih odgovora „Ostalo” kako bi korisnik mogao i dalje ispuniti web-obrazac, iako nove aplikacije nema među ponuđenim odgovorima. Kada korisnik odabere opciju „Ostalo”, prikazuje mu se novo pitanje „Unesite ime aplikacije:” u kojemu treba unijeti ime nove aplikacije za koju je uočio neispravnost. U budućnosti će možda postojati potreba za nekim drugim pitanjem kako bi saznali sve potrebne informacije za ispravak greške. Zbog toga je važno omogućiti implementaciju novih pitanja u postojeću web-aplikaciju. Tako nije potrebno izraditi novu web-aplikaciju nego je moguće nadograditi postojeću po potrebi. Novo pitanje može se implementirati ovako: `label` za postavljanje pitanja `<label>Pitanje </label>`. Za dodavanje zvjezdice koja vizualno označava da je pitanje obavezno, uz novo pitanje postavlja se zvjezdica `<label>Pitanje *</label>`.

Ovisno o vrsti odgovora može se staviti tekstualno polje, radio gumb, padajući izbornik, polje za unos više-rednog teksta ili potvrdni okvir.

U sljedećim primjerima prikazana je implementacija najčešće korištenih vrsta odgovora za ovu web-aplikaciju (Kôd 7.1, Kôd 7.2, Kôd 7.3, Kôd 7.4).

Tekstualno polje implementirano je na način da `input` polje sadrži `type="text"`.

```
<input type="text" value={inputValue}
onChange={handleChange}/>
```

Kôd 7.1 – Implementacija tekstualnog polja

Radio gumbi implementirani su pomoću input polja sa type="radio" za svaki ponuđeni radio gumb.

```
<div>
  <input type="radio" value="opcija1" checked={selectedOption
  === "opcija1"} onChange={handleChange}/>
  <label>Ime opcije1</label>
  <input type="radio" value="opcija2" checked={selectedOption
  === "opcija2"} onChange={handleChange}/>
  <label>Ime opcije2</label>
</div>
```

Kôd 7.2 – Implementacija radio gumba

Padajući izbornik implementiran je pomoću select u kojem se nalaze opcije option sa različitim vrijednostima value="vrijednost1".

```
<select value={selectedOption} onChange={handleChange}>
  <option value="">Odaberite opciju</option>
  <option value="opcija1">Opcija 1</option>
  <option value="opcija2">Opcija 2</option>
  <option value="opcija3">Opcija 3</option>
</select>
```

Kôd 7.3 – Implementacija padajućeg izbornika

Ako se želi postaviti obavezno polje, doda se required unutar koda za taj odgovor.

Prilagodba poruke obaveznog pitanja vrši se tako da se doda prilagođena poruka s
onInvalid={(e)=>e.target.setCustomValidity('Prilagođena poruka')}

i ukloni poruka kada je unesen odgovor s

onInput={(e)=>e.target.setCustomValidity('')}.

```
<input type="text" value={inputValue}
  onChange={handleChange}required onInvalid={(e) =>
  e.target.setCustomValidity('Prilagođena poruka')}
  onInput={(e) => e.target.setCustomValidity('')}/>
```

Kôd 7.4 – Primjer implementacija kada je pitanje obavezno i ima prilagođenu poruku

Zbog osiguravanja dugoročne funkcionalnosti potrebno je osigurati održavanje web-aplikacije. Imajući na umu tehnološki napredak, predviđen je razvoj novih aplikacija i za njih se osigurao unos. Uz to je omogućena i nadogradnja web-aplikacije kako bi u

budućnosti bilo moguće unijeti nove komponente bez potrebe za izradom potpuno nove web-aplikacije.

Zaključak

Cilj ovoga rada bio je razviti web-aplikaciju za prijavu neispravnosti u ICT-AAC aplikacijama te pregled poslanih prijava korištenjem tehnologije *React*.

Prvo su analizom postojećih rješenja i potreba korisnika, uočeni najčešći odabiri i potrebe. Na temelju tih podataka, definirani su programski zahtjevi web-aplikacije te je na kraju razvijena web-aplikacija. Korisnicima je sada omogućeno da kada u aplikaciji pronađu greške, mogu otići na ovu web-aplikaciju te jednostavno i intuitivno prijaviti grešku putem web-obrasca. Nakon toga, osobe koje su zadužene za ispravak grešaka mogu pregledati sve poslane web-obrasce te ispraviti greške.

Tijekom razvoja korištene su suvremene tehnologije kao što je *React* za korisničko sučelje, *Express.js* za instalaciju i pokretanje lokalnog poslužitelja te XML datoteka za pohranu podataka. Web-aplikacija nudi mnoge funkcionalnosti: unos podataka, prilaganje datoteke, validaciju podataka, prikaz poruka ako obavezna polja nisu ispunjena te prijenos, upis i prikaz podataka.

U budućnosti će možda postojati potreba za nadogradnjom i poboljšanjem. Primjerice, možda će biti potrebno dodati nove ponuđene odgovore ili dodati nova pitanja. Zbog toga je omogućena nadogradnja web-aplikacije, a u poglavlju „potrebe održavanja” opisan je taj proces.

Razvojem ove web-aplikacije za praćenje korisničkih prijava neispravnosti, postignuti su zadane ciljevi. U sklopu ovoga rada korisnicima se omogućilo da lakše prijave neispravnosti u ICT-AAC aplikacijama putem web-aplikacije. Pregled prijava osobama koje ispravljaju greške olakšava i ubrzava rad. Uz to, podaci prikupljeni analizom podataka mogu biti podloga za buduća istraživanja ili razvoj novih web-aplikacija.

Literatura

- [1] Politika privatnosti ICT-AAC aplikacija, ICT-AAC. <http://www.ict-aac.hr/index.php/hr/politika-privatnosti>; pristupljeno 3.6.2024.
- [2] Razvijene aplikacije, ICT-AAC. <http://www.ict-aac.hr/index.php/hr/ict-aac-razvijene-aplikacije>; pristupljeno 3.6.2024.
- [3] Prijava grešaka na ICT-AAC aplikacijana, Jotform. Poveznica: <https://form.jotforme.com/91832141751352>; pristupljeno 14. ožujka 2024.
- [4] Home, ICT-AAC. Poveznica: <http://www.ict-aac.hr/index.php/hr/>; pristupljeno 27. svibnja 2024.
- [5] Your first form, Mdn web docs. Poveznica: https://developer.mozilla.org/en-US/docs/Learn/Forms/Your_first_form; pristupljeno 1.6.2024.
- [6] Kristen Baker, Web Forms: The Ultimate Guide, Hubspot, (2022, rujan). Poveznica: <https://blog.hubspot.com/marketing/web-forms>; pristupljeno 1.6.2024.
- [7] How can we help, Jotform. Poveznica: <https://www.jotform.com/contact/>; pristupljeno 14. ožujka 2024.
- [8] Kontakti, Zagrebački električki tramvaj. Poveznica: <https://www.zet.hr/kontakti-276/276>; pristupljeno 14. ožujka 2024.
- [9] Pomoć, Gov.hr. Poveznica: <https://gov.hr/hr/pomoc/22>; pristupljeno 14 ožujka.2024.
- [10] Kontakti, Fina. Poveznica: <https://www.fina.hr/kontakti>; pristupljeno 14. ožujka 2024.
- [11] Help, HRTi SelfCare. Poveznica: <https://hrti-selfcare.hrt.hr/pages/help>; pristupljeno 14. ožujka 2024.
- [12] Kontakt, Tisak paket. Poveznica: <https://www.tisakpaket.hr/hr-HR/kontakt.html>; pristupljeno 14. ožujka 2024.
- [13] Kontaktirajte nas, Konzum. Poveznica: <https://www.konzum.hr/korisnicka-podrska/kontakt>; pristupljeno 14. ožujka 2024.
- [14] Web obrazac za prijavu stanja brojila, Brodo-plin. Poveznica: <https://www.brod-plin.hr/web-obrazac-za-prijavu-stanja-brojila/>; pristupljeno 18. ožujka 2024.
- [15] Obrazac sustava za dojavu i praćenje uhvaćenih, usmrćenih, ozlijeđenih i bolesnih strogo zaštićenih životinja, ArcGIS Survey123. Poveznica: <https://survey123.arcgis.com/share/894e91f99c3540038be13baf979ed800>; pristupljeno 18. ožujka 2024.
- [16] What is React, W3schools. Poveznica: https://www.w3schools.com/whatis/whatis_react.asp; pristupljeno 20. svibnja 2024.
- [17] Matija Koren, Psihologija boja: kako utječu na našu percepciju, emocije i svakodnevni odabir, Studentski.hr, (2020, listopad). Poveznica: <https://studentski.hr/zabava/zanimljivosti/psihologija-boja-kako-utjecu-na-nasu-percepciju-emocije-i-svakodnevni-odabir>; pristupljeno 12. svibnja 2024

- [18] Dan Abramov, Supported Browsers and Features, Create React App, (2021, srpanj). Poveznica: <https://create-react-app.dev/docs/supported-browsers-features/>; pristupljeno 4. svibnja 2024.
- [19] Stunning royalty-free images & royalty-free stock, Pixabay. Poveznica: <https://pixabay.com/>; pristupljeno 7. travnja 2024.

Sažetak

Web-aplikacija za praćenje korisničkih prijava neispravnosti temeljena na tehnologiji React

U ovom završnom radu opisuje se razvoj web-aplikacije za praćenje korisničkih prijava neispravnosti pronađenih u aplikacijama koje se koriste za komunikaciju i edukaciju ICT-AAC. Svrha aplikacije je omogućiti korisnicima da mogu jednostavno i intuitivno prijaviti neispravnost na koju su naišli kako bi se ona mogla ispraviti. U radu je prikazan razvoj i implementacija web-aplikacije. Analizirana su postojeća slična rješenja i potrebe korisnika kako bi se definirali programski zahtjevi. U raspravi je opisan proces testiranja aplikacije te su elaborirane potrebe održavanja. Za razvoj web-aplikacije korištene su tehnologije React i Express.js kako bi web-aplikacija imala intuitivno korisničko sučelje te se omogućilo učinkovito upravljanje podacima.

Ključne riječi: web-aplikacija, prijava neispravnosti, praćenje prijava, React, ICT-AAC

Summary

A web application for tracking user-detected failures based on React technology

This thesis describes the process of developing a web application for tracking user-detected failures in ICT-AAC web applications used for communication and education sector. The purpose of this web application is to ensure that the users can simply and intuitively report failures they came across so that these failures can then be corrected. The thesis outlines the development and the implementation of the web application. It contains an analysis of the existing solutions and user needs to define the technical requirements. The discussion describes the testing process and elaborates on the maintenance needs of the web application. The technologies React and Express.js were used for the development of the web application so that the user interface is intuitive, and the data management is efficient.

Keywords: web application, bug reporting, issue tracking, React, ICT-AAC

Skraćenice

ICT-AAC Information and Communication Technology based Augmentative and Alternative Communication potpomognuta komunikacija temeljena na informacijsko komunikacijskim tehnologijama

XML Extensible Markup Language proširiv jezik za označavanje

DOM Document Object Model model objekata dokumenata

CSS Cascading Style Sheets stilski jezik