

Proširenje i unaprjeđenje AudIT sustava za prikupljanje povratne informacije od publike na predavanju

Kovačević, Danijel

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:168:569970>

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

Download date / Datum preuzimanja: **2025-04-01**



Repository / Repozitorij:

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



SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1336

**PROŠIRENJE I UNAPRJEĐENJE AUDIT SUSTAVA ZA
PRIKUPLJANJE POVRATNE INFORMACIJE OD PUBLIKE NA
PREDAVANJU**

Danijel Kovačević

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1336

**PROŠIRENJE I UNAPRJEĐENJE AUDIT SUSTAVA ZA
PRIKUPLJANJE POVRATNE INFORMACIJE OD PUBLIKE NA
PREDAVANJU**

Danijel Kovačević

Zagreb, lipanj 2024.

ZAVRŠNI ZADATAK br. 1336

Pristupnik: **Danijel Kovačević (0036533868)**
Studij: Elektrotehnika i informacijska tehnologija i Računarstvo
Modul: Računarstvo
Mentor: doc. dr. sc. Juraj Petrović

Zadatak: **Proširenje i unaprjeđenje AudIT sustava za prikupljanje povratne informacije od publike na predavanju**

Opis zadatka:

U okviru završnog rada potrebno je proširiti AudIT sustav za prikupljanje povratne informacije od publike na predavanju (<https://audit.altii.online>) s barem dva dodatna tipa pitanja te dodatnim mogućnostima za predavača koje uključuju barem unos točnog odgovora na pitanje i kanala za izravnu komunikaciju s publikom. Također je potrebno unaprijediti dohvaćanje podataka za prikaz na sučelju predavača kroz server-sent events tehnologiju.

Rok za predaju rada: 14. lipnja 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1336

**Proširenje i unaprjeđenje AudIT sustava za
prikupljanje povratne informacije od publike
na predavanju**

Danijel Kovačević

Zagreb, rujan 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1336

**Proširenje i unaprjeđenje AudIT sustava za
prikupljanje povratne informacije od publike
na predavanju**

Danijel Kovačević

Zagreb, rujan 2024.

SVEUČILIŠTE U ZAGREBU

FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Zagreb, 4. ožujka 2024.

ZAVRŠNI ZADATAK br. 1336

Pristupnik: **Danijel Kovačević (0036533868)**

Studij: Elektrotehnika i informacijska tehnologija i Računarstvo

Modul: Računarstvo

Mentor: doc. dr. sc. Juraj Petrović

Zadatak: **Proširenje i unaprjeđenje AudIT sustava za prikupljanje povratne informacije od publike na predavanju**

Opis zadatka:

U okviru završnog rada potrebno je proširiti AudIT sustav za prikupljanje povratne informacije od publike na predavanju (<https://audit.altii.online>) s barem dva dodatna tipa pitanja te dodatnim mogućnostima za predavača koje uključuju barem unos točnog odgovora na pitanje i kanala za izravnu komunikaciju s publikom. Također je potrebno unaprijediti dohvaćanje podataka za prikaz na sučelju predavača kroz server-sent events tehnologiju.

Rok za predaju rada: 14. lipnja 2024.

Zahvaljujem se doc. dr. sc. Jurju Petroviću na pomoći i strpljenju. Zahvaljujem se svima koji su mi bili potpora za završetak studija.

Sadržaj

Uvod	1
1. Općenito o sustavu AudIT	2
1.1. AudIT u ulozi predavača	2
1.2. AudIT u ulozi publike	6
2. Server-sent events tehnologija.....	9
2.1. Arhitektura server-sent events tehnologije.....	9
2.1.1. Poslužiteljska strana	10
2.1.2. Klijentska strana	11
2.2. Prednosti i nedostaci server-sent event tehnologije.....	12
2.3. Primjena server-sent event tehnologije	13
3. Proširenje i unaprjeđenje AudIT sustava.....	14
3.1. Opcija zahtjeva za točan odgovor.....	14
3.1.1. Omogućavanje opcije	14
3.1.2. Unos i prikaz točnog odgovora.....	16
3.1.3. Osjetljivost na velika i mala slova i točno podudaranje	17
3.2. Kanal za izravnu komunikaciju s publikom	19
3.2.1. Omogućavanje opcije	19
3.2.2. Izmjena poruka putem kanala za komunikaciju s publikom	21
3.3. Server-sent events na sučelju predavača	22
Zaključak	23
Literatura	24
Sažetak.....	25
Summary.....	26

Uvod

AudIT je sustav za prikupljanje povratne informacije od publike na predavanjima od strane predavača [1]. Namijenjen je korištenju od strane predavača i publike. Funkcionira na način da predavač postavi neko pitanje, publika daje odgovore na njega te onda te dobivene odgovore predavač može vidjeti te ih prikazati i analizirati na razne načine.

Cilj je ovog rada proširiti sustav AudIT te dodati neke nove funkcionalnosti i mogućnosti. Potrebno je dodati nove mogućnosti za predavača, odnosno implementirati funkcionalnosti omogućavanja unosa točnog odgovora na pitanje te kanala za izravnu komunikaciju s publikom, koji omogućava publici jednostavno slanje poruka na sučelje predavača. Također je potrebno omogućiti drugačije dohvaćanje podataka za prikaz na sučelju predavača upotrebom SSE (server-sent events) tehnologije.

1. Općenito o sustavu Audit

Audit je alat koji služi za prikupljanje povratne informacije od publike te prikaz te povratne informacije na sučelju predavača. Namijenjen je korištenju na predavanjima, što uživo, što virtualnim, ali i bilo kojim drugim događajima gdje je potrebno prikupiti povratnu informaciju od publike, poput konferencija ili radionica. Sustav Audit koristi se u dvije uloge: uloga predavača i uloga publike. Na slici 1.1 prikazan je odabir uloge u sustavu.



Slika 1.1: Odabir uloge predavača ili publike, preuzeto iz [1]

1.1. Audit u ulozi predavača

Nakon što odabere tu ulogu, predavač na početku otvara sobu. To može biti nova soba, pri čemu unosi željeno ime sobe, željenu lozinku te e-mail (opcionalno). Na slici 1.2 prikazano je sučelje za otvaranje nove sobe.



Slika 1.2: Otvaranje nove sobe, preuzeto iz [1]

Druga opcija za otvaranje jest aktivna soba koja je napravljena ranije, pri čemu predavač unosi ime sobe koju želi reaktivirati te njenu lozinku, što je prikazano na slici 1.3.



Slika 1.3: Reaktiviranje sobe, preuzeto iz [1]

Nakon otvaranja sobe, predavaču se prikazuje glavno sučelje sobe, prikazano na slici 1.4. S lijeve strane nalaze se općenite postavke sobe. Slijedi izbornik za promjenu tipa pitanja, zatim dijelovi sučelja koji nam omogućavaju primanje odgovora kao i mogućnosti za podešavanje broja dozvoljenih te broja ponuđenih odgovora. Ispod toga nalaze se različite

moгуćnosti prikaza podataka, odgovora i korisnika te izbornik za promjenu tipa prikaza. Na slici 1.4 kao tip prikaza podešen je stupčasti graf, a postoji još i mogućnost prikaza tortnim grafom.



Slika 1.4: Prikaz sućelja za predavaća, preuzeto iz [1]

Postavke sobe uključuju sljedeće mogućnosti:

- **Resetiraj trenutno pitanje** – opcija za brzo resetiranje svih postavki trenutnog pitanja te mogućnost ponovnog podešavanja tipa pitanja te drugih svojstava pitanja
- **Popis korisnika** – opcija za odabir i učitavanje datoteke s korisničkim ID-jevima i imenima
- **AUDIT redirect** – opcija za korištenje AUDIT redirect koja omogućava preusmjeravanje tekstualnih poruka publike u neku drugu aplikaciju
- **Preuzmi podatke** – opcija preuzimanja podataka poslanih od strane publike u obliku CSV datoteke
- **Obriši podatke** – opcija koja nudi brisanje pojedinih podataka sobe, poput odgovora te podataka o korisnicima, ali i same sobe
- **Algoritam grupiranja** – opcija za automatsko grupiranje pojedinih tekstualnih odgovora na temelju sličnosti
- **Jezik** – opcija za promjenu jezika, dostupni su hrvatski, engleski, slovenski te estonski jezik

Na slici 1.5 prikazane su postavke sobe.

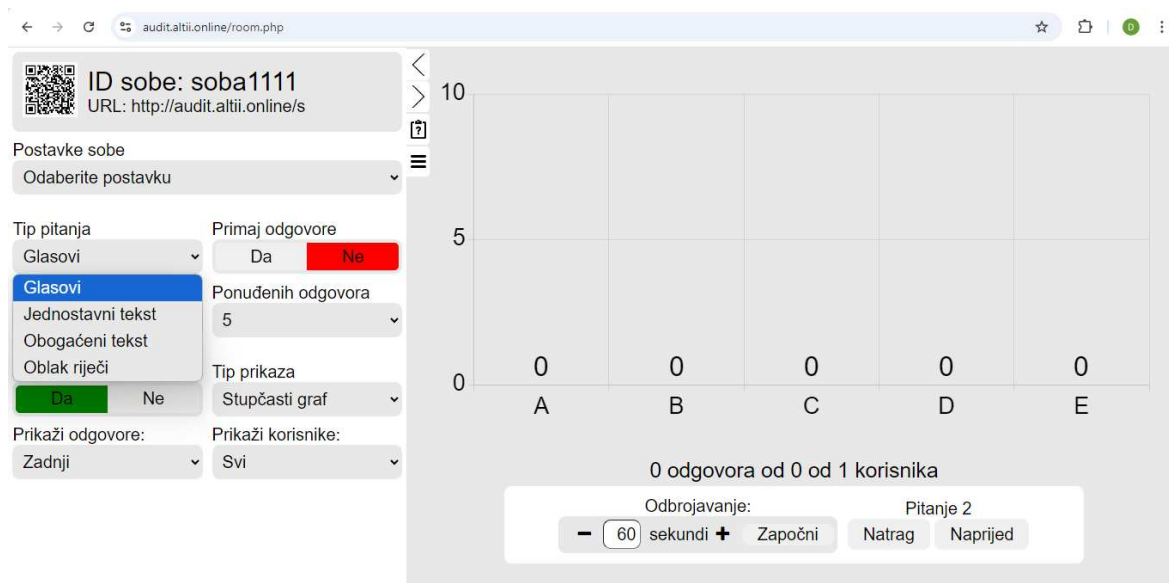


Slika 1.5: Popis postavki sobe, preuzeto iz [1]

Postoje četiri tipa pitanja, a to su:

- **Glasovi** – pitanje s ponuđenim odgovorima (ABC-pitanja)
- **Jednostavni tekst** – pitanje s odgovorima u tekstualnom obliku bez dodatnog formatiranja
- **Obogaćeni tekst** – pitanje s odgovorima prilagođenim za matematičke formule, indekse, eksponente i slično
- **Oblak riječi** – pitanje s odgovorima u tekstualnom obliku prikazanim kao oblak riječi

Na slici 1.6 prikazane su opcije za četiri različita tipa pitanja.

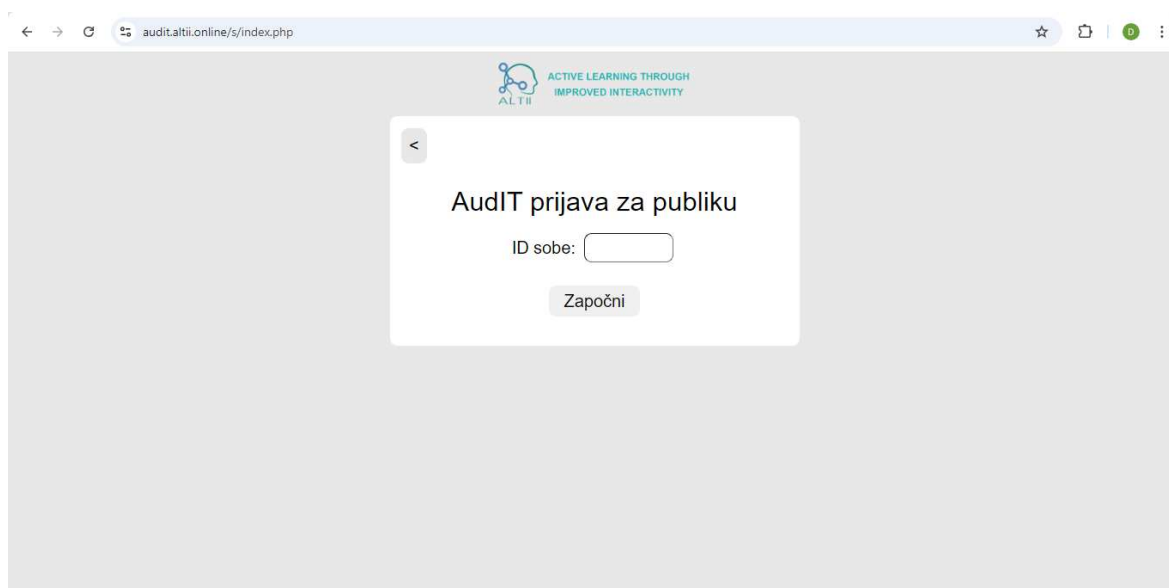


Slika 1.6: Tipovi pitanja, preuzeto iz [1]

Nakon što se svi potrebni podaci prikupe, predavač ima mogućnost prelaska na prethodno ili sljedeće pitanje.

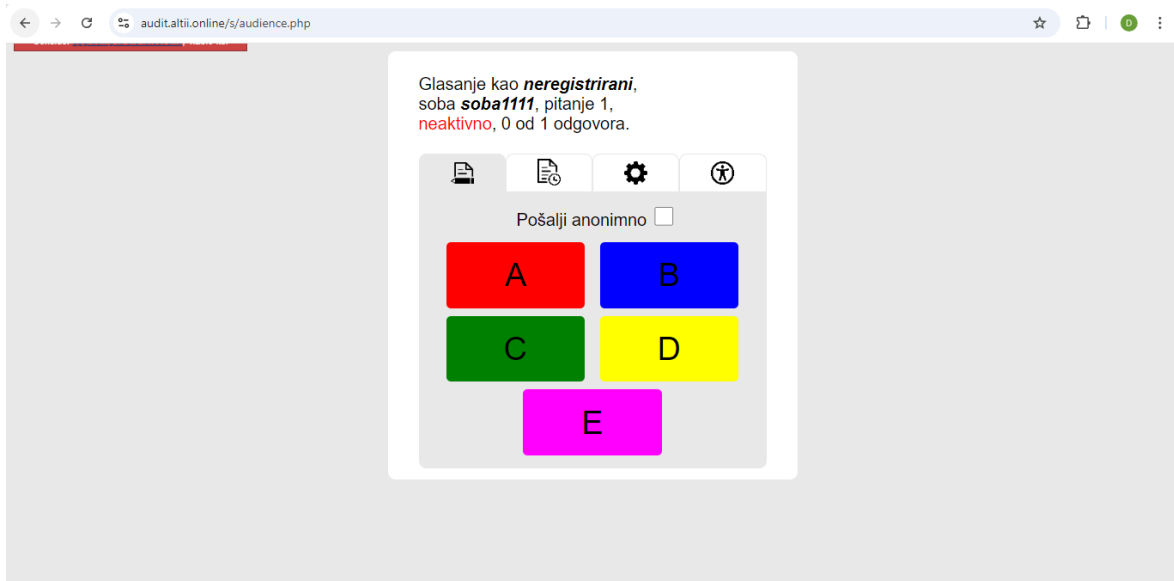
1.2. AudIT u ulozi publike

Osim predavača, druga uloga u kojoj se korisnik može prijaviti u aplikaciju jest uloga publike. Jedino što je potrebno za prijavu u ovoj ulozi jest ime sobe. Na slici 1.7 prikazano je sučelje za prijavu u ulozi publike.



Slika 1.7: Prijava za publiku, preuzeto iz [1]

Nakon unošenja imena sobe, prikazuje se sučelje za davanje odgovora, kao što je prikazano na slici 1.8.



Slika 1.8: Sučelje za davanje odgovora

Ono može varirati ovisno o tipu pitanja koji postavi predavač. Za ostale tipove pitanja ponuđen je tekstualni okvir za unos jednostavnog teksta, obogaćenog teksta ili oblaka riječi. Iznad toga nalaze se informacije o tome je li korisnik registriran ili neregistriran, ime sobe, broj pitanja, status aktivnosti označen od strane predavača te broj poslanih i mogućih odgovora. To je jedna od četiri opcije na sučelju za publiku. Preostale tri opcije uključuju pregled poslanih odgovora od strane korisnika, mogućnost registracije korisnika s korisničkim ID-jem te opcije vezane uz pristupačnost, koje uključuju povećanje i smanjenje slova, reset sučelja te visoki kontrast. Na slici 1.9 prikazana je opcija registracije u sustav.



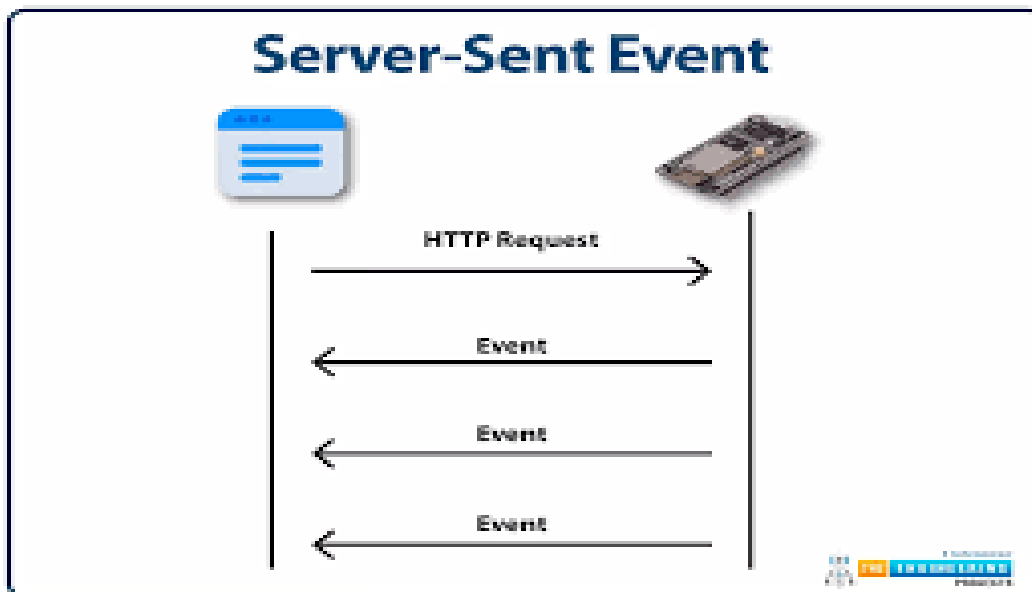
Slika 1.9: Opcija registracije za publiku, preuzeto iz [1]

2. Server-sent events tehnologija

Server-sent events (SSE) je poslužiteljska tehnologija koja klijentu omogućuje primanje automatskih ažuriranja od poslužitelja putem HTTP veze [2]. SSE funkcionira na način da se uspostavi početna klijentska veza između poslužitelja i klijenta, a nakon toga se podaci u stvarnom vremenu šalju od strane poslužitelja prema klijentu. Slanje podataka je jednosmjerno, dakle nije moguće slanje podataka od klijenta prema poslužitelju. Za razliku od tradicionalnog pristupa koji koristi AJAX za periodično dohvaćanje podataka, SSE omogućuje poslužitelju da proaktivno šalje ažuriranja klijentu kada su novi podaci dostupni. Zbog toga je SSE učinkovitije i bolje rješenje za aplikacije koje zahtijevaju sinkronizaciju podataka u stvarnom vremenu, kao što su aplikacije za čavrljanje, vijesti, sportski rezultati i sl. Ova tehnologija podržava automatsko ponovno povezivanje i jednostavno rukovanje podacima na klijentskoj strani.

2.1. Arhitektura server-sent events tehnologije

SSE tehnologija temelji se na HTTP protokolu. Koristi *text/event-stream* kao tip sadržaja za uspostavljanje kontinuirane veze između poslužitelja i klijenta. Ova veza ostaje otvorena, što omogućuje poslužitelju da šalje ažuriranja klijentu kada postanu dostupna. Za razliku od WebSocketeta, čija je veza dvosmjerna, SSE omogućava isključivo jednosmjernu komunikaciju, u smjeru poslužitelj-klijent. Proces se odvija na način da klijent inicira vezu s poslužiteljem, a zatim poslužitelj klijentu šalje podatke. Primjer tog procesa, gdje se klijent nalazi na lijevoj, a poslužitelj na desnoj strani, ilustriran je na slici 2.1



Slika 2.1: Prikaz arhitekture u SSE tehnologiji, preuzeto iz [3]

2.1.1. Poslužiteljska strana

Tip sadržaja koji koristi za uspostavljanje veze između poslužitelja i klijenta jest *text/event-stream* te se postavlja na poslužiteljskoj strani. Osim tipa prikaza, moguće je staviti i *cache-control*, koji označava hoće li se podaci kontrolirati u predmemoriji ili ne. Podaci se ispisuju metodom `echo`, a metoda `flush()` služi za ispis iz međuspremnika. Ispod se nalazi jednostavan prikaz koda poslužitelja u PHP-u.

```
<?php
header('Content-Type: text/event-stream');
header('Cache-Control: no-cache');

$time = date('r');
echo "data: The server time is: {$time}\n\n";
flush();
?>
```

Kôd 2.1 – Primjer poslužitelja u PHP-u, preuzet iz [4]

2.1.2. Klijentska strana

Na klijentskoj strani komunikacija se uspostavlja pomoću sučelja `EventSource`, koje definiira sve značajke koje upravljaju povezivanjem s poslužiteljem, primanjem događaja/podataka, pogreškama, zatvaranjem veze i tako dalje [5]. Za otvaranje veze prema poslužitelju kreira se novi `EventSource` objekt. Instanca `EventSource` otvara trajnu vezu s HTTP poslužiteljem, koji šalje događaje u formatu `text/event-stream`. Veza ostaje otvorena dok se ne zatvori pozivom `EventSource.close()` [6]. Kao argumenti se stavljaju link poslužitelja te opcija `withCredentials` koja označava hoće li se u sklopu HTTP zahtjeva slati vjerodajnice poput kolačića, HTTP autorizacije i slično. Ispod je prikazano jednostavno instanciranje sučelja `EventSource` [7].

```
const evtSource = new
EventSource("//api.example.com/ssedemo.php", {
  withCredentials: true,
});
```

Postoji nekoliko različitih tipova događaja koji mogu biti poslani. Tri su najvažnija: `onmessage`, `onopen` i `onerror` [4].

`Onmessage` se emitira svaki put kada `EventSource` primi poruku od poslužitelja. Koristi se za obradu podataka koje poslužitelj šalje klijentu. U kodu 1.2 prikazan je jednostavni primjer tog događaja.

```
const eventList = document.querySelector('ul');
evtSource.onmessage = (e) => {
  const newElement = document.createElement('li');
  newElement.textContent = 'message: ${e.data}';
  eventList.appendChild(newElement);
}
```

Kôd 2.2 – Primjer `onmessage` događaja, preuzet iz [6]

`Onopen` se pokreće prilikom otvaranja veze s izvorom događaja. Uglavnom se obrađuje na način da se ispiše prigodna poruka o otvaranju veze. Ispod je prikazan primjer tog događaja te rukovanja tim događajem [6].

```
evtSource.onopen = (e) => {
  console.log("The connection has been established.");
};
```

Onerror se pojavljuje prilikom greške u otvaranju veze. Generira se kada se pojave problemi poput isteka vremena ili problema s kontrolom pristupa. Ispod je prikazan primjer rukovanja tim događajem [7].

```
evtSource.onerror = (err) => {  
    console.error("EventSource failed: ", err);  
};
```

Prema zadanim postavkama, u slučaju prekida veze između poslužitelja i klijenta, veza se ponovno pokreće. Vezu se može prekinuti pozivom metode `EventSource.close()`.

2.2. Prednosti i nedostaci server-sent event tehnologije

Kao i svaka tehnologija, i server-sent event tehnologija ima svoje prednosti i nedostatke [8].

Glavne prednosti:

- Ponovno povezivanje – u slučaju prekida veze, veza se automatski ponovno pokreće
- Manje opterećenje – u usporedbi s WebSocketima, potrebno je manje resursa za održavanje komunikacije
- Nema blokiranja vatrozidom – za razliku od WebSoketa, server-sent event tehnologija ne susreće se s problemima u blokiranju vatrozidom

Glavni nedostaci:

- Ograničenje formata podataka – podržano je samo slanje UTF-8 poruka, dok prijenos binarnih podataka nije podržan
- Jednosmjernost – poruke je moguće slati samo u jednom smjeru, od strane poslužitelja prema strani klijenta
- Ograničenje broja istodobnih veza – za razliku od WebSoketa, SSE ima ograničenje od samo šest istodobnih veza prema nekom pregledniku

2.3. Primjena server-sent event tehnologije

Server-sent event tehnologija uglavnom se primjenjuje u aplikacijama koje sinkroniziraju podatke u stvarnom vremenu. U odnosu na WebSokete ima veću jednostavnost i pouzdanost te je prikladno rješenje kada je samo to dovoljno te kada ne treba komunikaciju u dva smjera, već samo jednom.

Uobičajeni slučajevi upotrebe server-sent event tehnologije uglavnom su aplikacije za vijesti, čavrljanje, rezultate sportskih događaja u stvarnom vremenu, praćenje stanja burzi u stvarnom vremenu, a koristi se i za prikaz obavijesti na društvenim mrežama i slično. Također se koristi u aplikacijama za praćenje u kojima poslužitelji trebaju slati upozorenja klijentima, poput aplikacija za nadzor mreže.

3. Proširenje i unaprijeđenje AudIT sustava

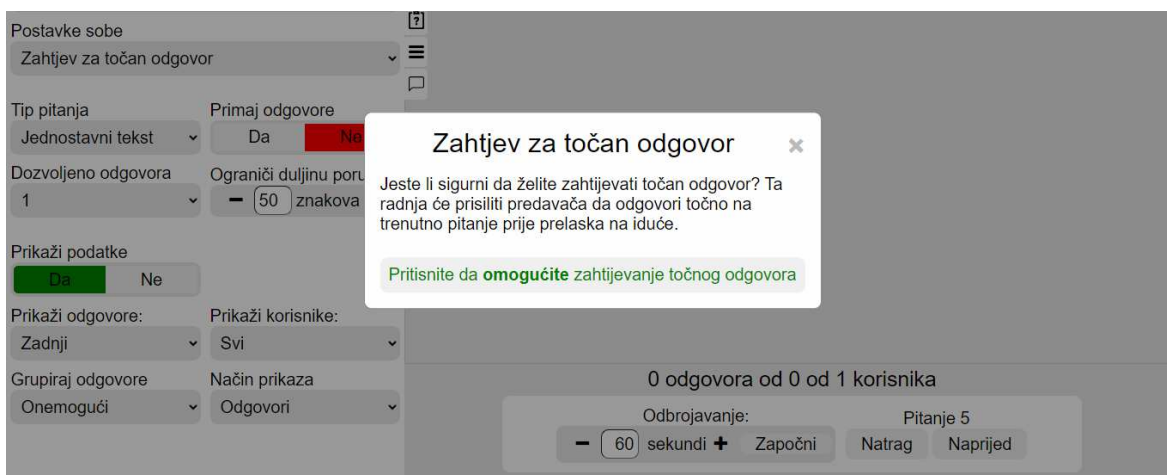
Glavni zadatak ovog rada bio je proširiti i unaprijediti sustav AudIT. Programski jezici koji se koriste jesu PHP i JavaScript. Prvi dio zadatka obuhvaćao je dodavanje funkcionalnosti koje uključuju unos točnog odgovora na pitanje i kanal za izravnu komunikaciju s publikom.

3.1. Opcija zahtjeva za točan odgovor

Unos točnog odgovora na pitanje funkcionalnost je čija je svrha omogućiti predavaču unos točnog odgovora na pitanje, dok publici daje povratnu informaciju o tome je li im odgovor točan.

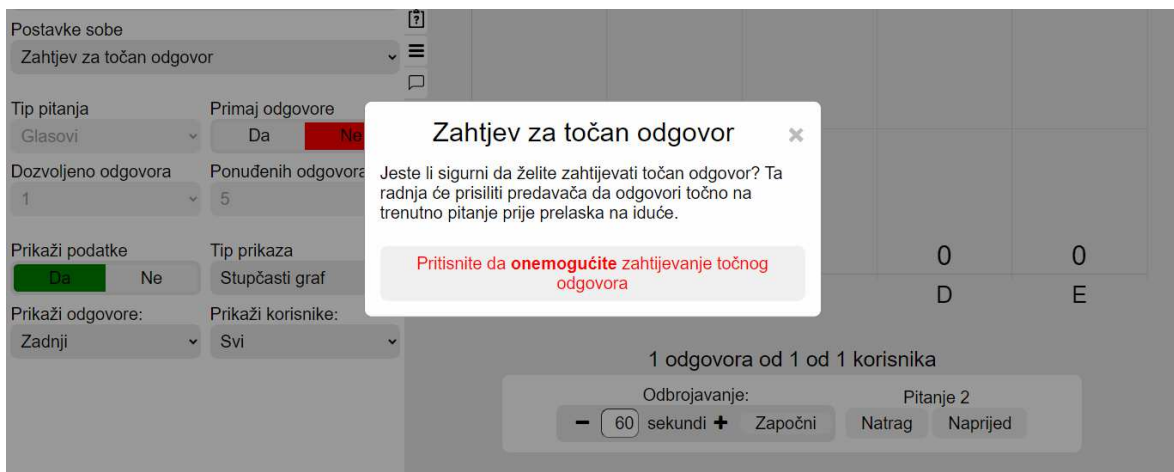
3.1.1. Omogućavanje opcije

Da bi koristio ovu mogućnost, predavač mora odabrati glavni izbornik te odabrati opciju *Zahtjev za točan odgovor*. Nakon toga, predavaču se nudi opcija da klikom na gumb omogući unošenje točnog odgovora na pitanje. Na slici 3.1 prikazana je ta opcija.



Slika 3.1: Prikaz opcije omogućavanja zahtijevanja točnog odgovora, preuzet iz [1]

U slučaju da je opcija zahtijevanja točnog odgovora već uključena, predavaču se nudi opcija da klikom na gumb onemogućí zahtijevanje unosa točnog odgovora. Ta situacija prikazana je na slici 3.2.



Slika 3.2: Prikaz opcije onemogućavanja zahtijevanja točnog odgovora, preuzet iz [1]

Na slici 3.3 prikazan je programski kod gdje je vidljivo na koji je način u kodu izvedena ova mogućnost.

```

<!-- Room option: Prompt for correct answer -->
<div id='promptForCorrectAnswer' class='modal-content'>
  <span class="close" onclick="document.getElementById('myModal').style.display='none';
  document.getElementById('roomOptionSelector').selectedIndex = 0;">&times;</span>
  <div class="modal-title"><?=$translate["Prompt for correct answer"] ?></div>
  <?=$translate["Prompt for correct answer message"]?><br /><br />
  <button class="interface_label modal-button"
  style="margin-top:5px; color: <?=$promptForCorrectAnswer ? 'red' : 'green' ?>;"
  onclick="configureRoomOptions('promptForCorrectAnswer', <?=$promptForCorrectAnswer ? 'false' : 'true' ?>);"
  onclick="configureQuestion('questionType', document.getElementById('chartSelector').value);"
  type="button" name="promptForCorrectAnswerButton"
  id="promptForCorrectAnswerButton"><?=$promptForCorrectAnswer ?
  $translate["Click to disable prompt for correct answer"] : $translate["Click to enable prompt for correct answer"] ?>
</button>
</div>

```

Slika 3.3: Programski kod za opciju zahtijevanja točnog odgovora, preuzet iz [1]

U prvih nekoliko linija koda nalaze se općenite stvari vezano uz *div* elemente. S obzirom na to da postoje dva različita prikaza ovog elementa, prikazana na slikama 3.1 i 3.2, zanimljivo je za izdvojiti korištenje ternarnih operatora. Ispod, u kodu 3.1, izdvojeni su dijelovi koda koji, s obzirom na vrijednost varijable `$promptForCorrectAnswer`, mijenjaju boju gumba te mijenjaju sam tekst gumba.

```

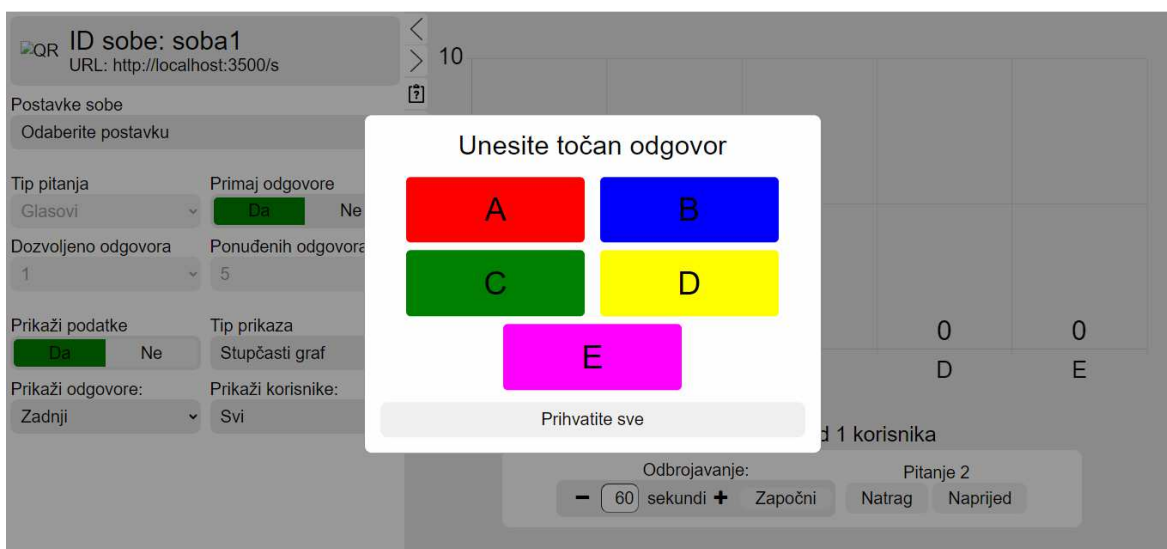
color: <?=$promptForCorrectAnswer ? 'red' : 'green' ?>;
<?=$promptForCorrectAnswer ?
$translate['Click to disable prompt for correct answer'] :
$translate['Click to enable prompt for correct answer'] ?>

```

Kôd 3.1 – Programski kod u PHP-u koji mijenja boju i tekst, preuzet iz [1]

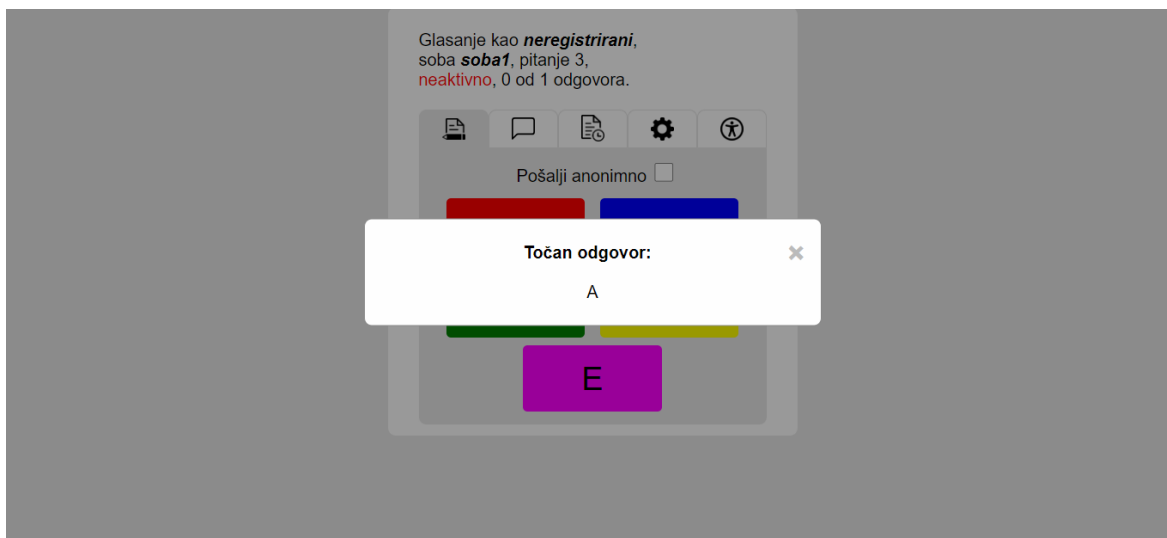
3.1.2. Unos i prikaz točnog odgovora

Ako se predavač odluči na opciju zahtijevanja točnog odgovora, nakon prikupljanja odgovora publike i stiskanja opcije za prelazak na iduće pitanje na njegovom se sučelju prikazuje opcija unosa točnog odgovora. Ona se mijenja s obzirom na vrstu pitanja. Na slici 3.4 prikazana je opcija unosa točnog odgovora za tip pitanja glasove.



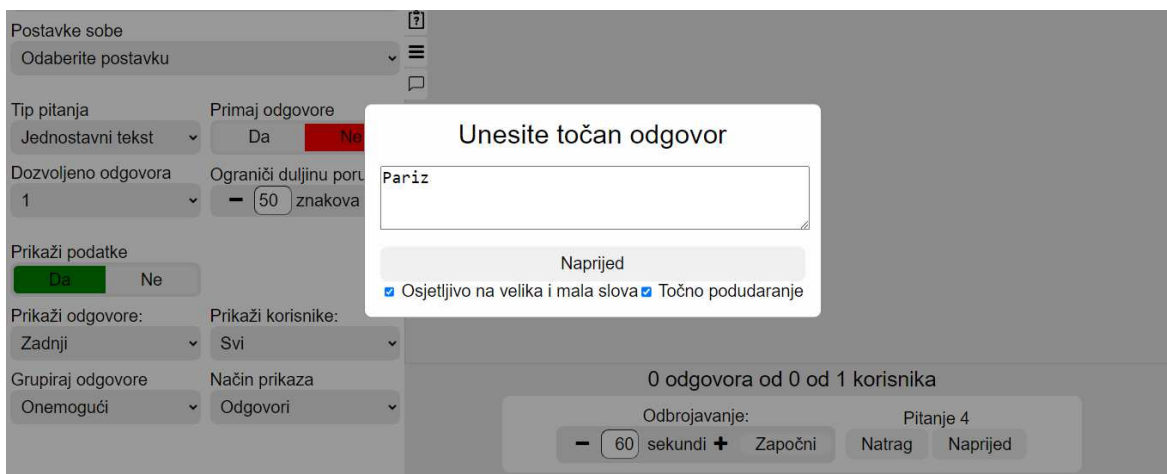
Slika 3.4: Prikaz opcije unosa točnog odgovora za glasove na sučelju predavača, preuzet iz [1]

Nakon što predavač odabere točan odgovor, na njegovom sučelju događa se prijelaz na sljedeće pitanje. Istovremeno, na sučelju publike javlja se informacija o točnom odgovoru. Ta je situacija prikazana na slici 3.5.



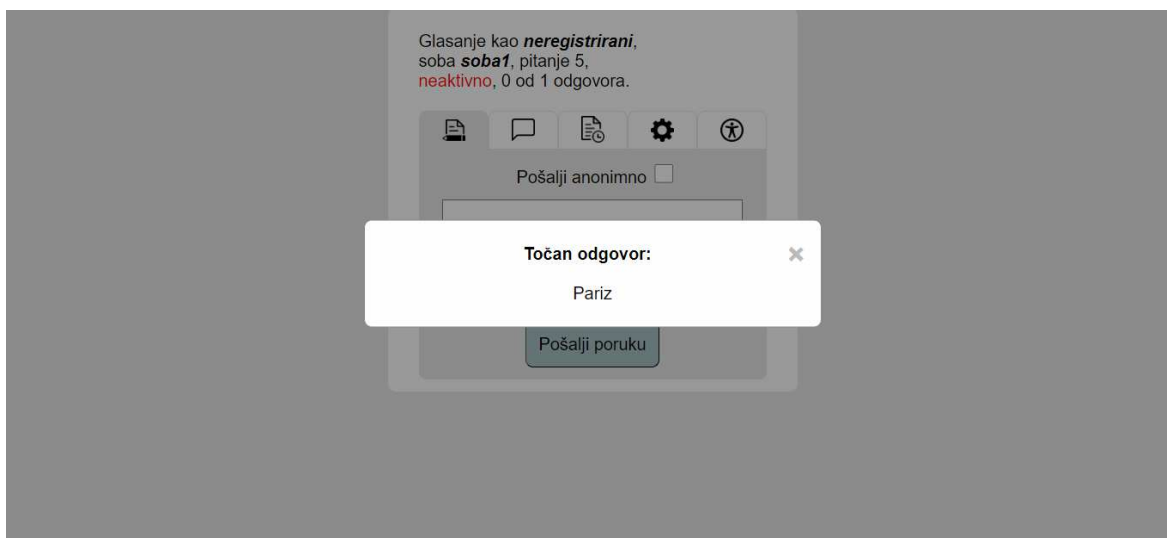
Slika 3.5: Prikaz točnog odgovora za glasove na sučelju publike, preuzet iz [1]

Za tipove pitanja oblak riječi i jednostavni tekst dodane su opcije *Exact match* te *Case sensitive*. Njihova je uloga provjera točnog podudaranja te osjetljivosti na velika i mala slova. Na slici 3.6 prikazana je opcija unosa točnog odgovora za jednostavni tekst.



Slika 3.6: Prikaz unosa točnog odgovora za jednostavni tekst na sučelju predavača, preuzet iz [1]

Isto kao i za glasove, predavač unosi točan odgovor te nakon toga prelazi na iduće pitanje. Na sučelju publike prikazuje se točan odgovor, što je prikazano na slici 3.7.



Slika 3.7: Prikaz točnog odgovora za jednostavni tekst na sučelju publike, preuzet iz [1]

3.1.3. Osjetljivost na velika i mala slova i točno podudaranje

Opcije osjetljivosti na velika i mala slova te točnog podudaranja bilo je potrebno dodati samo za jednostavni tekst i oblak riječi jer za glasove i obogaćeni tekst te opcije nemaju smisla. To je postignuto uvjetnim *checkboxom*, odnosno ispitivanjem vrijednosti koja

predstavlja tip pitanja. Dopusćene vrijednosti su 2, koja predstavlja jednostavni tekst te 4, koja predstavlja oblak rijeći. Te vrijednosti čitaju se iz datoteke `questionsConfigFile.csv`. Nakon ulaska u tu datoteku, čita se zadnji red, odnosno element na petoj poziciji u navedenoj datoteci. Potom se izvršava provjera radi li se o nekoj od dopuštenih vrijednosti. Ako da, prikazuju se navedeni *checkboxovi*, dok se u suprotnom ne događa ništa. U kodu 3.2 prikazana je ova implementacija.

```
<?php
    $allowedTypes = [2, 4];
    $csvFile =
        $GLOBALS['roomsDir'].$roomID.'/'. $GLOBALS['questionConfigFile'];
    if (($handle = fopen($csvFile, 'r')) !== FALSE) {
        $lastRow = null;
        while (($dataCSV = fgetcsv($handle, 1000, ',') !== FALSE) {
            $lastRow = $dataCSV;
        }
        fclose($handle);
        if in_array($lastRow[5], $allowedTypes) {
            echo '<input type="checkbox"
                name="exactMatchCheckbox"
                value="1">
                <label for="caseSensitiveCheckbox">' .
                $translate["Case sensitive"] .
                '</label>';

            echo '<input type="checkbox"
                name="exactMatchCheckbox"
                value="1">
                <label for="exactMatchCheckbox">' .
                $translate["Exact match"] .
                '</label>';
        }
    }
?>
```

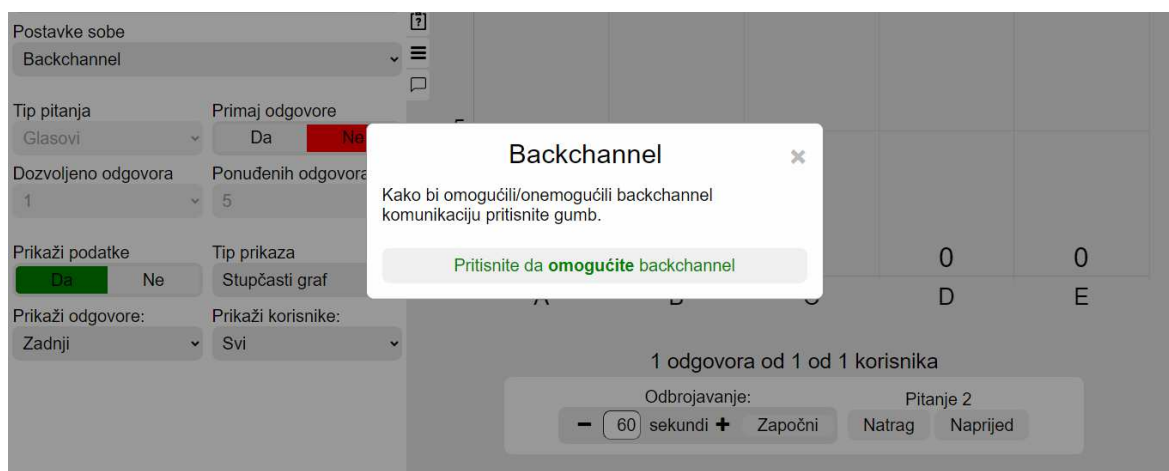
Kôd 3.2 – Programski kod u PHP-u koji implementira uvjetni *checkbox*, preuzet iz [1]

3.2. Kanal za izravnu komunikaciju s publikom

Ova funkcionalnost služi kako bi se mogla ostvariti komunikacija između predavača i publike tekstualnim porukama. Predavač ima mogućnost uključiti i isključiti tu opciju, a publika može slati poruke predavaču ako je opcija uključena.

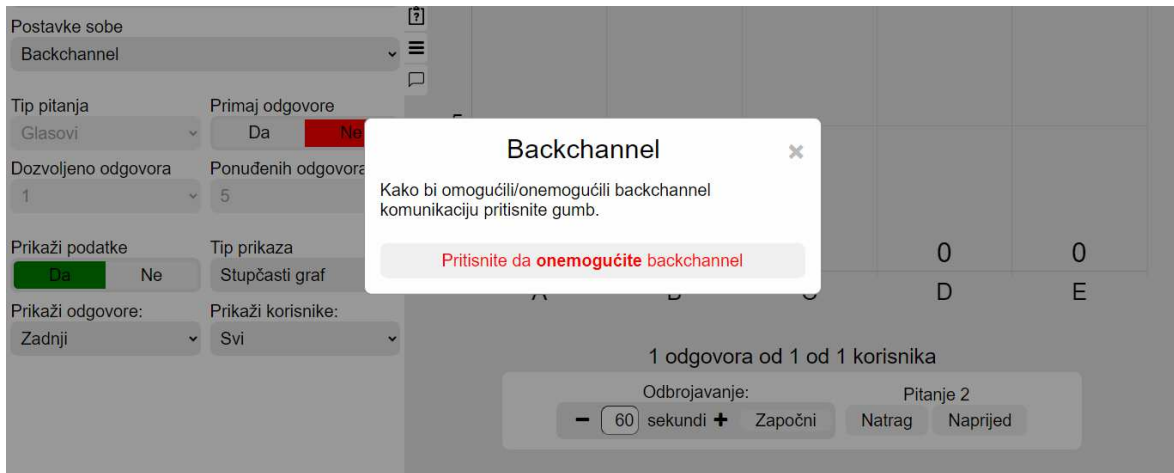
3.2.1. Omogućavanje opcije

Da bi koristio ovu mogućnost, predavač odabire glavni izbornik te odabire opciju *Backchannel*. Nakon toga, predavaču se nudi opcija za omogućavanje i onemogućavanje ove opcije, ovisno o tome što je trenutno odabrano. Na slici 3.8 prikazana je opcija omogućavanja *backchannela*.



Slika 3.8: Prikaz opcije omogućavanja *backchannela*, preuzet iz [1]

Ako je ova opcija već uključena, predavač je može isključiti ponovnim klikom na isti gumb. Ta je mogućnost prikazana na slici 3.9.



Slika 3.9: Prikaz opcije omogućavanja *backchannela*, preuzet iz [1]

Ovaj je dio izveden na sličan način kao i kod opcije za unos točnog odgovora, a na slici 3.10 nalazi se programski kod koji definira gore navedene opcije.

```

<!-- Room option: Backchannel -->
<div id='backchannel' class='modal-content'>
  <span class="close" onclick="document.getElementById('myModal').style.display='none';
  document.getElementById('roomOptionSelector').selectedIndex = 0;">&times;</span>
  <div class="modal-title"><?=$translate["Backchannel"] ?></div>
  <?=$translate["Backchannel message"]?><br /><br />
  <button class="interface_label modal-button" style="margin-top:5px; color: <?=$backchannel ? 'red' : 'green' ?>";
  onclick="configureRoomOptions('backchannel', <?!$backchannel?>)"
  type="button" name="backchannelButton" id="backchannelButton"><?=$backchannel ?
  $translate["Click to disable backchannel"] : $translate["Click to enable backchannel"] ?></button>
</div>

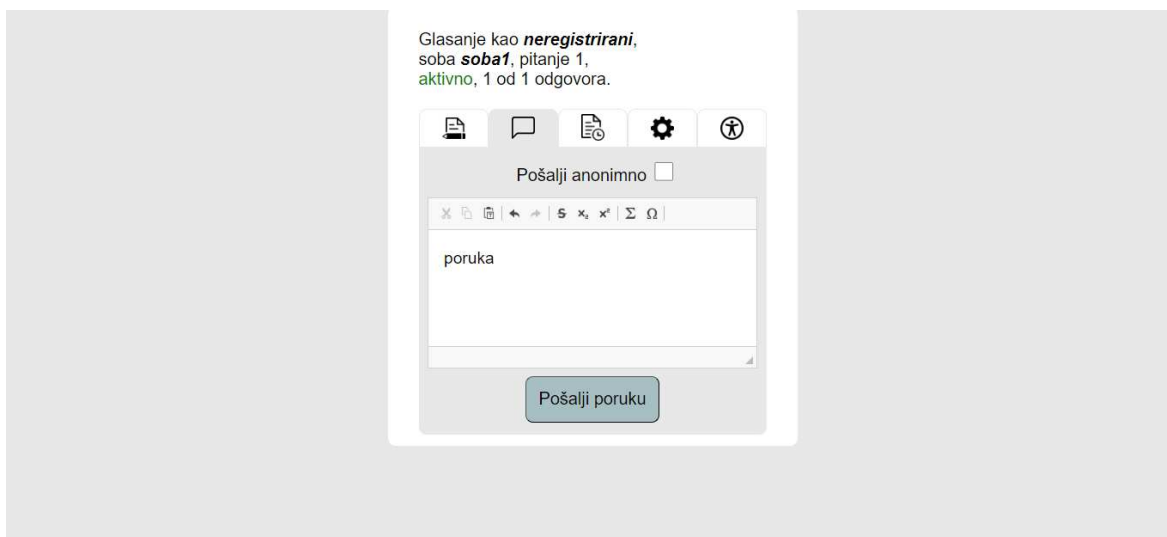
```

Slika 3.10: Programski kod za opciju omogućavanja *backchannela*, preuzet iz [1]

Kao i kod opcije za omogućavanje unosa točnog odgovora, i ovdje su dva različita prikaza omogućena korištenjem ternarnog operatora uz provjeru varijable `$backchannel`.

3.2.2. Izmjena poruka putem kanala za komunikaciju s publikom

Nakon što predavač omogući *backchannel*, na sučelju za publike pojavit će se mogućnost korištenja kanala za komunikaciju. Klikom na gumb otvara se okvir u koji publika može upisati svoju poruku za predavača te ju poslati. Mogućnost slanja poruke prikazana je na slici 3.11.



Slika 3.11: Prikaz slanja poruke publike prema predavaču na sučelju publike, preuzet iz [1]

Nakon što netko iz publike pošalje poruku, ona se prikazuje na sučelju predavača. Poruka se također istovremeno sprema u datoteku `messages.csv`, a osim same poruke, šalju se i podaci poput datuma, vremena, korisničkog imena pošiljatelja i slično. Na slici 3.12 na lijevoj strani prikazane su poruke na sučelju predavača pristigle od strane publike.



Slika 3.12: Prikaz primljenih poruka na sučelju predavača, preuzet iz [1]

3.3. Server-sent events na sučelju predavača

U ovom dijelu zadatka bilo je potrebno unaprijediti dohvaćanje podataka na sučelju predavača pomoću server-sent events tehnologije. Dohvat podataka na sučelju predavača bio je izveden na način da se funkcija `refreshDisplay()` poziva u intervalima po svake dvije sekunde te se unutar te funkcije formiraju „GET“ HTTP-zahtjevi koji se šalju na `getData.php`. Taj dio koda, gdje se događa poziv funkcije, prikazan je u nastavku.

```
function startRefresh() {
    refresher = setInterval(function(){refreshDisplay();},2000);
}
```

Takvo rješenje nije najzgodnije zbog prevelikog broja zahtjeva koji se šalju. Umjesto toga, formirana je datoteka `dataChange.php` koja predstavlja poslužitelja u server-sent event tehnologiji.

Ona funkcionira na način da provjerava jesu li se podaci promijenili. Ako je došlo do promjene podataka, podatke treba poslati metodom `flush()`, a u suprotnom ne radi ništa, odnosno spava tri sekunde pomoću metode `sleep()`. Provjera je li došlo do promjene podataka odvija se na način da se provjeri jesu li trenutno spremljeni podaci istovjetni onima dobivenima. Kod te provjere prikazan je ispod.

```
if ($previousVotes == votes/N.csv) && ($previousText == text/N.csv)
```

Veza između servera i klijenta uspostavlja se pomoću sučelja `EventSource()`. Kao parametri se predaju ID sobe te `timeout` koji iznosi devedeset minuta. Način na koji se to obavlja prikazan je u odsječku koda ispod.

```
var source = new EventSource('dataChange.php?room=?php echo $roomID
?>&timeout=90');
```

Klijent u SSE-tehnologiji predstavlja datoteka `room.php`. Njena funkcija je parsirati podatke poslana sa servera, što se odvija u dijelu `onmessage`. Nakon što se to obavi, podaci se prikazuju na način istovjetan kao i u prethodnom rješenju.

Zaključak

U ovome radu provedeno je proširenje i unaprjeđenje AudIT sustava za prikupljanje povratne informacije publike. Na početku rada dan je kratak uvod u AudIT sustav te je objašnjeno što je dosad napravljeno u njemu te na koji način funkcionira. Prikazana je teoretska pozadina server-sent events tehnologije, primjeri korištenja te prednosti i mane u usporedbi s drugim sličnim tehnologijama.

U praktičnom dijelu rada implementirano je proširenje sustava dvama dodatnim mogućnostima za predavača: zahtijevanje unosa točnog odgovora na pitanje te kanal za izravnu komunikaciju s publikom. Dohvaćanje podataka za prikaz na sučelju predavača unaprijeđeno je pomoću server-sent events tehnologije. Na slikama su prikazivani dijelovi sustava bitni za rad te ključni dijelovi koda.

Literatura

- [1] AUDIT – Audience response system. Poveznica: <https://audit.altii.online/>; pristupljeno 31. kolovoza 2024.
- [2] Wikipedia. Server-sent events. Poveznica: https://en.wikipedia.org/wiki/Server-sent_events; pristupljeno 31. kolovoza 2024.
- [3] The Engineering Projects. Server-Sent Events with ESP32 and DHT11. Poveznica: <https://www.theengineeringprojects.com/2022/03/server-sent-events-with-esp32-and-dht11.html>; pristupljeno 1. rujna 2024.
- [4] W3Schools. HTML Server-Sent Events API. Poveznica: https://www.w3schools.com/html/html5_serversentevents.asp; pristupljeno 1. rujna 2024.
- [5] Mozilla Developer Network. Server-Sent Events – Web APIs. Poveznica: https://developer.mozilla.org/en-US/docs/Web/API/Server-sent_events; pristupljeno 1. rujna 2024.
- [6] Mozilla Developer Network. EventSource. Poveznica: <https://developer.mozilla.org/en-US/docs/Web/API/EventSource>; pristupljeno 1. rujna 2024.
- [7] Mozilla Developer Network. Using Server-Sent Events – Web APIs. Poveznica: https://developer.mozilla.org/en-US/docs/Web/API/Server-sent_events/Using_server-sent_events; pristupljeno 1. rujna 2024.
- [8] Ably. WebSockets vs Server-Sent Events. Poveznica: <https://ably.com/blog/websockets-vs-sse>; pristupljeno 1. rujna 2024.

Sažetak

Proširenje i unaprjeđenje AudIT sustava za prikupljanje povratne informacije od publike na predavanju

AudIT je sustav napravljen za interakciju predavača i publike te prikupljanje povratne informacije od publike. U ovom radu sustav AudIT ukratko je predstavljen i opisan te je proširen dodatnim mogućnostima za predavača. Te mogućnosti uključuju zahtijevanje unosa točnog odgovora na pitanje te uspostavu kanala za izravnu komunikaciju između publike i predavača. Također je provedena analiza server-sent events tehnologije te je napravljeno unaprjeđenje dohvata podataka za prikaz na sučelju predavača pomoću te tehnologije.

Ključne riječi: AudIT, server-sent events

Summary

Enhancing and Improving the AudIT Audience Response System

AudIT is a system made for lecturer-audience interaction and gathering feedback from the audience. In this paper, the AudIT system is briefly presented and described, and it is expanded with additional possibilities for the lecturer. These possibilities include requiring the entry of the correct answer to the question and establishing a channel for direct communication between the audience and the lecturer. An analysis of the server-sent events technology was also carried out, and an improvement was made in retrieving data for display on the lecturer's interface using this technology.

Keywords: AudIT, server-sent events