

Aplikacija za rezervaciju učionica

Student:

Miloš Petrović ITS 07/23

Sadržaj

1. Uvod.....	3
2. Opis Projekta.....	3
2.1 Ciljevi Projekta	3
2.2 Funkcionalnosti.....	4
3. Tehnologije	4
4. Opis projekta kroz back-end tehnologije	5
4.1 PHP deo:	5
4.2 MySql – baza deo:	8
5. Korisnički Interfejs	10
6. Benefiti.....	11
7. Izazovi.....	11
7.1 Šta je CORS?	11
8. Zaključak.....	14

1. Uvod

Tokom prakse, radio sam na razvoju web aplikacije za rezervaciju učionica na fakultetu koristeći React za frontend i PHP za backend. Cilj aplikacije je omogućiti profesorima da pregledaju dostupnost učionica i rezervišu termine za predavanja ili druge aktivnosti.

2. Opis Projekta

Aplikacija za zakazivanje učionica razvijena je kao praktično rešenje za upravljanje rezervacijama i terminima korišćenja učionica u obrazovnim ustanovama. Glavni cilj ovog projekta je da pojednostavi proces rezervacije učionica, smanji konflikte oko zakazivanja i obezbedi transparentnost i efikasnost u korišćenju prostora.

2.1 Ciljevi Projekta

Primarni ciljevi projekta su:

- Omogućiti korisnicima (profesorima, studentima, administrativnom osoblju) jednostavan način za pregled dostupnosti učionica.
- Pružiti mogućnost rezervacije učionica na jednostavan i intuitivan način.
- Automatski izbegavati konflikte u zakazivanju i obaveštavati korisnike o eventualnim preklapanjima.
- Osigurati transparentnost u korišćenju resursa kroz jasan prikaz svih rezervacija.

2.2 Funkcionalnosti

Aplikacija za zakazivanje učionica obuhvata sledeće ključne funkcionalnosti:

- **Prijava i Autentifikacija:** Korisnici mogu da se prijave putem jednostavnog interfejsa. Sistem autentifikacije obezbeđuje da samo ovlašćeni korisnici mogu da vrše rezervacije.
- **Pregled Dostupnosti:** Interaktivni kalendar omogućava korisnicima da pregledaju dostupnost učionica u realnom vremenu.
- **Zakazivanje Termina:** Korisnici mogu da rezervišu učionice za određene termine.
- **Administrativni Pregled i Upravljanje:** Administratori imaju poseban pristup za pregled i upravljanje.

3. Tehnologije

Frontend: React, CSS

Backend: PHP, MySQL

4. Opis projekta kroz back-end tehnologije

Back-end sastoji od nekoliko ključnih delova:

4.1 PHP deo:

- intervals.php: Ovaj fajl je odgovoran za manipulaciju podacima vezanim za zauzete i slobodne termine učionica.

```
$workingHoursStart = 8;
$workingHoursEnd = 21;

$freeIntervals = array();
$lastEndTime = $workingHoursStart;

foreach ($busyIntervals as $interval) {
    $startTime = (int)$interval['startTime'];
    $endTime = (int)$interval['endTime'];

    if ($lastEndTime < $startTime) {
        $freeIntervals[] = array(
            'startTime' => $lastEndTime,
            'endTime' => $startTime
        );
    }
    $lastEndTime = max($lastEndTime, $endTime);
}

if ($lastEndTime < $workingHoursEnd) {
    $freeIntervals[] = array(
        'startTime' => $lastEndTime,
        'endTime' => $workingHoursEnd
    );
}
```

Slika 1: php kod za proveru zauzetih/slobodnih intervala

- `scheduled_appointments.php`: Ovaj fajl omogućava dobijanje zakazanih termina za određenog profesora na osnovu sesije.

```
if (isset($_SESSION['name'])) {  
    $professor_name = $_SESSION['name'];  
  
    $query = "SELECT * FROM busy_classrooms WHERE professor = '$professor_name'";  
    $result = query($query);  
  
    confirm($result);  
  
    $appointments = fetch_all($result);  
  
    echo json_encode($appointments);  
} else {  
    echo json_encode(["success" => false, "message" => "Niste ulogovani."]);  
}
```

Slika 2: dobavljanje zauzetih termina

- `busy_insert.php`: Ovaj fajl omogućava dodavanje novih zauzetih termina učionica u bazu podataka

```
$sql = "INSERT INTO busy_classrooms (calendar, startTime, endTime, amphitheater, professor)  
VALUES (:calendar, :startTime, :endTime, :amphitheater, :professor)";
```

Slika 3: SQL upit za unos zakazivanja

- `check_classrooms.php`: Ovaj fajl ažurira status zauzetosti učionica i briše prošle unose iz baze podataka na osnovu trenutnog vremena i datuma.

```
$sql = "SELECT amphitheater, startTime, endTime, professor  
FROM busy_classrooms  
WHERE calendar = '$currentDate'  
AND startTime <= '$currentTime'  
AND endTime > '$currentTime';  
$result = $conn->query($sql);
```

Slika 4: SQL upit za prikaz/ažuriranje zauzetih prostorija

- `add_professor.php`: Ovaj fajl omogućava dodavanje novih profesora u bazu podataka

```
$name = isset($data['name']) ? $data['name'] : null;
$surname = isset($data['surname']) ? $data['surname'] : null;
$email = isset($data['email']) ? $data['email'] : null;
$password = isset($data['password']) ? password_hash($data['password'], PASSWORD_BCRYPT) : null;

if ($name && $surname && $email && $password) {
    $query = "INSERT INTO professors (name, surname, email, password) VALUES (?, ?, ?, ?)";
    $stmt = $con->prepare($query);
    $stmt->bind_param("ssss", $name, $surname, $email, $password);

    if ($stmt->execute()) {
        echo json_encode(["success" => true, "message" => "Profesor uspešno dodat"]);
    } else {
        echo json_encode(["success" => false, "message" => "Greška prilikom dodavanja profesora"]);
    }

    $stmt->close();
} else {
    echo json_encode(["success" => false, "message" => "Nevažeći unos"]);
}
```

Slika 5: php kod za dodavanje profesora

- delete_professor.php: Ovaj fajl omogućava brisanje profesora iz baze podataka na osnovu primljenog ID-ja profesora.

```
if ($_SERVER['REQUEST_METHOD'] === 'DELETE') {
    if (isset($_GET['id'])) {
        $id = $_GET['id'];

        $sql = "DELETE FROM professors WHERE id = ?";
        if ($stmt = $con->prepare($sql)) {
            $stmt->bind_param("i", $id);
            if ($stmt->execute()) {
                echo json_encode(['success' => true, 'message' => 'Profesor uspešno obrisan']);
            } else {
                echo json_encode(['success' => false, 'message' => 'Greška pri brisanju profesora']);
            }
        }
    }
}
```

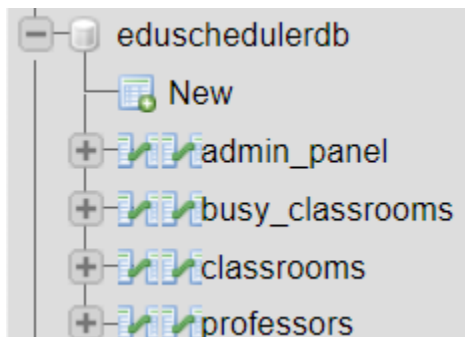
Slika 6: php kod za brisanje profesora

- update_professor.php: Ovaj fajl omogućava ažuriranje podataka o profesoru u bazi podataka

```
$stmt = $con->prepare("UPDATE professors SET name = ?, surname = ?, email = ? WHERE id = ?");
$stmt->bind_param("sssi", $name, $surname, $email, $id);
$stmt->execute();
```

Slika 7: php kod za izmenu kredicionala profesora

4.2 MySQL – baza deo:



Slika 8: Izgled baze sa tabelama

Prikaz dela strukture baze podataka pod nazivom eduschedulerrdb. Ova baza podataka sadrži nekoliko tabela koje se koriste za upravljanje rasporedom.

1. **admin_panel:**

- Ova tabela sadrži podatke vezane za administratore u aplikaciji.

id	username	password
1	admin	\$2y\$10\$1Ac9XfEVSNhDY0/ONwHX2eUVSUIExphChdYKy04t.jh...

Slika 9 Sadržaj tabele admin_panel

2. **busy_classrooms:**

- Ova tabela sadrži informacije o zauzetim učionicama.

amphitheater	startTime	endTime	professor	reason	calendar
A1	14	16	Aleksandra		2024-07-15
U7	8	10	Aleksandra		2024-07-17
A3	16	18	Dušan		2024-07-23
A2	12	14	Dušan		2024-07-29
U106	13	15	Slađana		2024-07-30
U207	8	12	Slađana		2024-08-20

Slika 10: Sadržaj tabele busy_classrooms

3. classrooms:

- Ova tabela sadrži osnovne informacije o svim učionicama.

id	classroom	occupied	features	floor
1	U1	0	Projector	1
2	U2	0	Projector, Computer Lab	1
3	U3	0	Lab Equipment, Projector	1
4	U4	0	Projector, Whiteboard	1
5	U5	0	Seminar Room, Projector	1
6	U6	0	Computer Lab, Projector	1
7	U7	0	Whiteboard, Seminar Room	1
8	U8	0	Projector, Whiteboard	1
9	U9	0	Projector, Computer Lab	1
10	A1	0	Lab Equipment, Projector	1
11	A2	0	Projector, Whiteboard	2
12	A3	0	Projector, Computer Lab	2

Slika 11: Sadržaj tabele classrooms

4. professors:

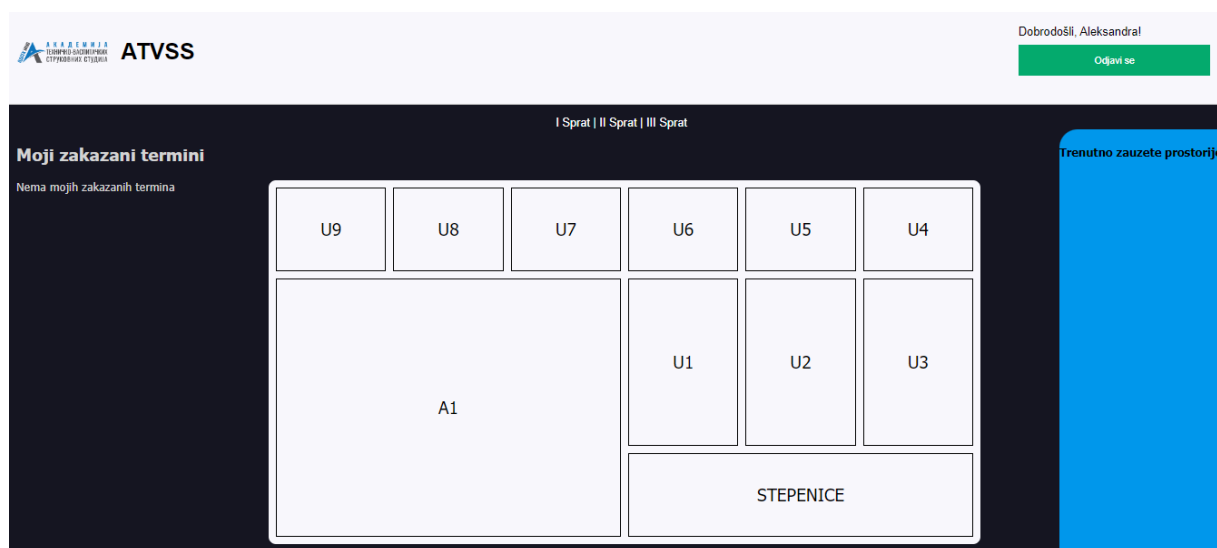
- Ova tabela sadrži informacije o profesorima.

id	name	surname	email	password
1	Aleksandra	Boričić	aleksandra.boricic@akademijanis.edu.rs	\$2y\$10\$4jbgA54LMhS8vAvZMaN3tOu46RPKllo8oLHTIfqU6Kp...
2	Aleksandra	Marinković	aleksandra.marinkovic@akademijanis.edu.rs	\$2y\$10\$17nd9.HbEiShBAuVfNuRuORohLjvm4bSp6nQJmzwJFu...
3	Anica	Milošević	anica.milosevic@akademijanis.edu.rs	\$2y\$10\$7l0gNWkS.tTASh9T4kBD8.9LNj6OaTNay55.YSgJ.DH...
4	Biljana	Milutinović	biljana.milutinovic@akademijanis.edu.rs	\$2y\$10\$5EadOwKtFfSs1l3ULAUykuFRiQKFCyvgLC0VHpYjHw6...
5	Boban	Cvetanović	boban.cvetanovic@akademijanis.edu.rs	\$2y\$10\$03sjhFmQ3tvX6lRnEKsxcOaxim0QcNNgiewLmlh.VU5...
6	Danijela	Zlatković	danijela.zlatkovic@akademijanis.edu.rs	\$2y\$10\$dRjg9XqLnKN4sh/g1FylQ.EGfhn7iF5qlmgTAvXBIT7...
7	Dejan	Blagojević	dejan.blagojevic@akademijanis.edu.rs	\$2y\$10\$Y6hTxKUDRcFm/OFONTkzjuEJWfIMXUpPeQ5sU99lhje...
8	Dejan	Bogičević	dejan.bogicevic@akademijanis.edu.rs	\$2y\$10\$mm.GcF/UHW.sNx/ZaRL1HedQuKln5EIZwc.348RwCJw...
9	Dušan	Radosavljević	dusan.radosavljevic@akademijanis.edu.rs	\$2y\$10\$oe7DPQ4BsIB.rk1q5J3uKes2IV/2sCQXFMbbmh2qk7c...
10	Dušan	Stefanović	dusan.stefanovic@akademijanis.edu.rs	\$2y\$10\$StLJAR1nKkPpq2Xs.4xllheer5LTyj1loP.53oHegtX4...

Slika 12: Sadržaj tabele professors

5. Korisnički Interfejs

Poseban akcenat stavljen je na intuitivan i user-friendly korisnički interfejs. Korisničko iskustvo je optimizovano tako da omogući lako snalaženje i brzu rezervaciju učionica. Interaktivni kalendar, jasno označene dostupne i zauzete učionice, kao i jednostavan sistem za prijavu i autentifikaciju čine ovu aplikaciju efikasnom i lako upotrebljivom.



Slika 13: korisnički interfejs



Slika 14: admin korisnički interfejs

6. Benefiti

Implementacija ove aplikacije donosi brojne benefite:

- Povećana Efikasnost: Brže i jednostavnije rezervisanje učionica smanjuje administrativne troškove i povećava efikasnost.
- Smanjenje Konflikata: Automatska kontrola preklapanja termina smanjuje mogućnost konflikata.
- Transparentnost: Jasna evidencija svih rezervacija omogućava bolju organizaciju i planiranje.

7. Izazovi

Jedan od novih izazova sa kojim se do sada nismo sretali je omogućavanje komunikacije između dva različita porta. Kao što je već navedeno, ova aplikacija je razvijena korišćenjem React.js biblioteke kao front-end alat koji radi na portu 3000, dok back-end radi na portu 80. Da bi se komunikacija između ova dva porta odvijala nesmetano, potrebno je dodati CORS politiku koja omogućava određene metode za rad između ova dva porta.

7.1 Šta je CORS?

CORS (Cross-Origin Resource Sharing) je sigurnosni mehanizam implementiran u web pretraživačima koji omogućava ili ograničava resurse koji se dele između različitih domena. U kontekstu web aplikacija, to znači da aplikacija koja se pokreće na jednom domenu (ili portu) može zahtevati resurse (kao što su API podaci) sa drugog domena. Bez pravilno postavljene CORS politike, pretraživači će blokirati ove zahteve iz sigurnosnih razloga. CORS politika definiše koje domene i koje HTTP metode su dozvoljene za pristup resursima na serveru, čime se obezbeđuje bezbedna razmena podataka između različitih izvora. [slika 15]

```
if (isset($_SERVER['HTTP_ORIGIN'])) {  
    header("Access-Control-Allow-Origin: {$_SERVER['HTTP_ORIGIN']}");  
    header('Access-Control-Allow-Credentials: true');  
    header('Access-Control-Max-Age: 86400'); // cache for 1 day  
}  
  
if ($_SERVER['REQUEST_METHOD'] == 'OPTIONS') {  
    if (isset($_SERVER['HTTP_ACCESS_CONTROL_REQUEST_METHOD']))  
        header("Access-Control-Allow-Methods: GET, POST, OPTIONS");  
  
    if (isset($_SERVER['HTTP_ACCESS_CONTROL_REQUEST_HEADERS']))  
        header("Access-Control-Allow-Headers: {$_SERVER['HTTP_ACCESS_CONTROL_REQUEST_HEADERS']}");  
  
    exit(0);  
}
```

Slika 15: Kod za CORS policy

Na slici je prikazan PHP kod koji implementira CORS (Cross-Origin Resource Sharing) zaglavlja za HTTP zahteve.

1. Prvi if uslov je proverava za HTTP_ORIGIN zaglavlje:

- Ako je postavljeno HTTP_ORIGIN zaglavlje (koje označava domen sa kojeg potiče zahtev), postavlja se Access-Control-Allow-Origin zaglavlje da dozvoli taj domen.
 - HTTP_ORIGIN zaglavlje je deo HTTP zahteva koji sadrži informaciju o originalnom izvoru (domeni) sa kojeg je zahtev poslat. Ovaj header je ključan za CORS (Cross-Origin Resource Sharing) mehanizam, koji omogućava web aplikacijama na jednom domenu da pristupaju resursima na drugom domenu.
- Access-Control-Allow-Credentials: true omogućava da se šalju kolačići i HTTP autentifikacija sa zahtevima.
- Access-Control-Max-Age: 86400 postavlja keširanje CORS odgovora na jedan dan (86400 sekundi).

2. Drugi if uslov je provera za HTTP metodu:

- Ako je metoda HTTP zahteva OPTIONS, proverava se postojanje HTTP_ACCESS_CONTROL_REQUEST_METHOD zaglavlja i postavlja Access-Control-Allow-Methods: GET, POST, OPTIONS da dozvoli navedene HTTP metode.
- Ako je postavljeno HTTP_ACCESS_CONTROL_REQUEST_HEADERS zaglavlje, postavlja se Access-Control-Allow-Headers sa vrednostima navedenih zaglavlja koja klijent zahteva.
- exit(0) zaustavlja dalje izvršavanje koda nakon slanja odgovora na OPTIONS zahtev.

8. Zaključak

Projekat aplikacije za zakazivanje učionica predstavlja praktično i moderno rešenje za upravljanje obrazovnim resursima. Njegova implementacija može značajno doprineti efikasnosti, organizaciji i zadovoljstvu korisnika u obrazovnim ustanovama.

Praksa mi je omogućila da steknem praktično iskustvo u razvoju web aplikacija koristeći React i PHP. Naučio sam kako efikasno upravljati stanjem aplikacije, kako raditi sa backend servisima i kako kreirati interaktivne i korisne interfejse. Ovaj projekat je bio izazovan, ali izuzetno koristan za moj dalji profesionalni razvoj.