

5 dana u oblacima 2025 – Challenge faza

Opis funkcionalnog zahteva

Studentski život je ispunjen obavezama, predavanjima, projektima i ispitima, svaki minut je dragocen. Upravo zato, studentske menze, koje bi trebalo da budu oaza brzog i pristupačnog obroka, često postaju mesto frustracije. Dugi redovi, gužve u vreme ručka, i večito pitanje “da li ću uopšte stići da jedem pre sledećeg predavanja?” - problemi su koje poznaje svaki student.

Kako bi se unapredilo iskustvo obedovanja i optimizovalo vreme provedeno u menzama, Studentski centar je kontaktirao firmu Levi9 s idejom razvoja inovativnog digitalnog rešenja - sistema za rezervaciju termina za obedovanje u studentskim menzama.

Cilj ovog sistema je da studentima omogući da unapred rezervišu termin za doručak, ručak ili večeru u željenoj menzi. Na taj način studenti mogu planirati svoje obroke, izbeći gužve i smanjiti vreme čekanja.

Vaš zadatak je da smanjite redove u menzama i studentima obezbedite lakši dolazak do obroka.

Opis zadatka

Sistem treba da prati raspoloživost menzi po vremenskim slotovima i obezbedi jednostavnu rezervaciju termina za obrok. Jedini korisnici ovog sistema treba da budu studenti.

Student treba da ima ime, prezime i jedinstvenu email adresu. **Student** dodatno može da bude redar (admin). Samo studenti koji su redari mogu da kreiraju menze i da vrše promene nad samim menzama. Ostali studenti koriste sistem za rezervaciju obroka.

Pored **Student** entiteta, treba implementirati i entitet **Canteen** odnosno menza. **Canteen** kao entitet predstavlja studentsku menzu koja pored svog jedinstvenog imena i lokacije sadrži i informacije o svom kapacitetu i terminima u kojima servira obroke (doručak, ručak i večera).

Poslednji entitet koji treba implementirati je **Reservation**. Za određenu menzu u periodu kada ona servira obroke i nije dostigla pun kapacitet, student može rezervisati svoj termin za obrok. Termin mora krenuti na pun sat ili na pola sata (primer 15:00 ili 15:30) i može biti dužine 30 ili 60 min. Nije dozvoljeno kreirati rezervaciju za dane u prošlosti. Svaki student može da kreira najviše jednu rezervaciju u izabranom vremenskom intervalu, bez obzira na

menzu u kojoj je rezervacija napravljena. Potrebno je obezbediti podršku za otkazivanje kreirane rezervacije ukoliko se student predomisli. Otkazana rezervacija treba da ostane vidljiva u sistemu sa statusom „Cancelled“.

Dodatno, **Student** redar može da ažurira podatke o menzi. U Challenge fazi zadatka nije potrebno voditi računa da li ove izmene utiču na ostale studente i već kreirane rezervacije. Takođe, **Student** redar treba da ima mogućnost brisanja menze koja prestaje sa radom. Ovom akcijom treba otkazati kreirane rezervacije za menzu koja je obrisana. Ni za akciju brisanja menze u ovoj fazi takmičenja nije potrebno voditi računa da li ove izmene utiču na ostale studente i već kreirane rezervacije.

Student treba da omogući i dohvaćanje informacija o svim menzama. Takođe, **Student** može da dohvati podatke o pojedinačnoj menzi i može da ispita kapacitet menze za zadati kriterijum. Taj kriterijum je definisan početnim i krajnjim datumom, vremenskim intervalom i trajanjem obroka (30 ili 60 min) za dane obuhvaćene tim datumima. Datumi definišu početni i krajnji dan, a vremena označavaju interval u toku jednog dana za koje je potrebno vratiti broj slobodnih mesta u trajanju od 30 ili 60 minuta (primer: za datume od 2025-11-21 do 2025-11-23 i vreme od 08:00 do 12:00, potrebno je vratiti za sva tri dana broj slobodnih mesta u intervalu od 08:00 do 12:00).

Detaljna definicija API-a je data u nastavku teksta.

Students API:

Kreiranje **Student**-a: **POST /students**

Request payload:

```
{
  "name": "Nikola Petrović",
  "email": "nikola.petrovic@etf.com",
  "isAdmin": false
}
```

Responses:

OK Response	
Status Code	201 Created
Body	{ "id": "1", "name": "Nikola Petrović", "email": "nikola.petrovic@etf.com", }

	"isAdmin": false }
Error Responses	
Status Code	4xx - Invalid input
Status Code	5xx - Server error

Dohvatanje **Student**-a po id-u: **GET /students/{{id}}**

Response:

OK Response	
Status Code	200 OK
Body	{ "id": "1", "name": "Nikola Petrović", "email": "nikola.petrovic@etf.com", "isAdmin": false }
Error Responses	
Status Code	404 Not Found

Canteens API:

Kreiranje menze: **POST /canteens**

Request payload:

Header	"studentId": "123456"
Body	{ "name": "Tri kostura", "location": "Obilićev venac 4, Beograd", "capacity": 100, "workingHours": [{ "meal": "breakfast", "from": "08:00", "to": "10:00" }, { "meal": "lunch", "from": "11:00", "to": "13:00" },] }

	<pre> "meal": "dinner", "from": "17:00", "to": "19:00" }] }</pre>
--	--

Response:

OK Response	
Status Code	201 Created
Body	<pre> { "id": "1", "name": "Tri kostura", "location": "Obilićev venac 4, Beograd", "capacity": 100, "workingHours": [{ "meal": "breakfast", "from": "08:00", "to": "10:00" }, { "meal": "lunch", "from": "11:00", "to": "13:00" }, { "meal": "dinner", "from": "17:00", "to": "19:00" }] }</pre>
Error Responses	
Status Code	4xx - Invalid input
Status Code	5xx - Server error

Dohvatanje **svih** menzi : **GET /canteens**

Response:

Status Code	200 OK
-------------	--------

Body	<pre>[{ "id": "1", "name": "Tri kostura", "location": "Obilićev venac 4, Beograd", "capacity": 100, "workingHours": [{ "meal": "breakfast", "from": "08:00", "to": "10:00" }, { "meal": "lunch", "from": "11:00", "to": "13:00" }, { "meal": "dinner", "from": "17:00", "to": "19:00" }] }, { "id": "2", "name": "Velika menza", "location": "Dr Sime Miloševića 4, Novi Sad", "capacity": 200, "workingHours": [{ "meal": "breakfast", "from": "07:00", "to": "10:00" }, { "meal": "lunch", "from": "12:00", "to": "15:00" }] }]</pre>
------	---

	<pre> { "meal": "dinner", "from": "17:00", "to": "20:00" }] }] </pre>
--	---

Dohvatanje informacija o pojedinačnoj menzi: **GET /canteens/{id}**

Response:

OK Response	
Status Code	200 OK
Body	<pre> { "id": "1", "name": "Tri kostura", "location": "Obilićev venac 4, Beograd", "capacity": 100, "workingHours": [{ "meal": "breakfast", "from": "08:00", "to": "10:00" }, { "meal": "lunch", "from": "11:00", "to": "13:00" }, { "meal": "dinner", "from": "17:00", "to": "19:00" }] } </pre>
Error Responses	
Status Code	404 Not Found

Ažuriranje menze: **PUT /canteens/{id}**

Request:

Header	"studentId": "123456"
Body	{ "name": "Tri kostura Updated" }

Response:

OK Response	
Status Code	200 OK
Body	{ "id": "1", "name": "Tri kostura Updated", "location": "Obilićev venac 4, Beograd", "capacity": 100, "workingHours": [{ "meal": "breakfast", "from": "08:00", "to": "10:00" }, { "meal": "lunch", "from": "11:00", "to": "13:00" }, { "meal": "dinner", "from": "17:00", "to": "19:00" }] }
Error Responses	
Status Code	404 Not Found
Status Code	4xx - Invalid input
Status Code	5xx - Server error

Brisanje menze: **DELETE /canteens/{id}**

Request:

Header	"studentId": "123456"
Body	{}

Response:

OK Response	
Status Code	204 No Content
Body	{}
Error Responses	
Status Code	404 Not Found
Status Code	4xx - Invalid input
Status Code	5xx - Server error

Dohvatanje preostalog kapaciteta u **svim** menzama za određeni vremenski period, željene vremenske intervale u toku dana i dužinu obroka: **GET** **/canteens/status?startDate={{date}}&endDate={{date}}&startTime={{time}}&endTime={{time}}&duration={{duration}}**

Response:

OK Response	
Status Code	200 OK
Body	[{ "canteenId": "1", "slots": [{ "date": "2025-10-31", "meal": "breakfast", "startTime": "09:00", "remainingCapacity": 30 }, { "date": "2025-10-31", "meal": "breakfast", "startTime": "09:30", "remainingCapacity": 15 },] }]

	<pre>"date": "2025-10-31", "meal": "lunch", "startTime": "11:00", "remainingCapacity": 22 }, { "date": "2025-10-31", "meal": "lunch", "startTime": "11:30", "remainingCapacity": 5 }] }, { "canteenId": "2", "slots": [{ "date": "2025-10-31", "meal": "breakfast", "startTime": "09:00", "remainingCapacity": 15 }, { "date": "2025-10-31", "meal": "breakfast", "startTime": "09:30", "remainingCapacity": 7 }, { "date": "2025-10-31", "meal": "lunch", "startTime": "11:00", "remainingCapacity": 16 }, { "date": "2025-10-31", "meal": "lunch", "startTime": "11:30",</pre>
--	--

	<pre> "remainingCapacity": 3 }] }] </pre>
Error Responses	
Status Code	4xx - Invalid input
Status Code	5xx - Server error

Dohvatanje preostalog kapaciteta u **jednoj** menzi po Id-u za određeni vremenski period, željene vremenske intervale u toku dana i dužinu obroka : **GET** **/canteens/{id}/status?startDate={{date}}&endDate={{date}}&startTime={{time}}&endTime={{time}}&duration={{duration}}**

Response:

OK Response	
Status Code	200 OK
Body	<pre> { "canteenId": "1", "slots": [{ "date": "2025-10-31", "meal": "breakfast", "startTime": "08:00", "remainingCapacity": 30 }, { "date": "2025-10-31", "meal": "breakfast", "startTime": "09:00", "remainingCapacity": 15 }, { "date": "2025-10-31", "meal": "lunch", "startTime": "11:00", </pre>

	<pre> "remainingCapacity": 22 }, { "date": "2025-10-31", "meal": "lunch", "startTime": "12:00", "remainingCapacity": 5 }] } </pre>
Error Responses	
Status Code	404 Not Found
Status Code	4xx - Invalid input
Status Code	5xx - Server error

Kreiranje rezervacije: **POST /reservations**

Request:

<pre> { "studentId": "123456", "canteenId": "2", "date": "2025-10-27", // today or in the future "time": "12:00", "duration": 30, // 30 or 60 } </pre>
--

Response:

OK Response	
Status Code	201 Created
	<pre> { "id": "1", "status": "Active", "studentId": "123456", "canteenId": "2", "date": "2025-10-27", "time": "12:00", "duration": 30 } </pre>

Error Responses	
Status Code	4xx - Invalid input
Status Code	5xx - Server error

Otkazivanje rezervacije: **DELETE /reservations/{id}**

Request:

Header	"studentId": "123456"
Body	{}

Response:

OK Response	
Status Code	200 OK
Body	{ "id": "1", "status": "Cancelled", "studentId": "123456", "canteenId": "2", "date": "2025-10-27", "time": "12:00", "duration": 30 }
Error Responses	
Status Code	404 Not Found
Status Code	4xx - Invalid input
Status Code	5xx - Server error

Tehnički zahtevi

Aplikacija može biti implementirana u jednoj od navedenih tehnologija: JavaScript/Typescript, Java, C#, Python.

Obavezno koristiti build alate/package manager-e, kao što su NPM/Yarn, Maven/Gradle, NuGet, Pip.

Obavezno je koristi in-memory bazu.

Rešenje mora u potpunosti biti zasnovano na open-source (free) software-u.

Izvorni kod rešenja mora biti upload-ovan na GitHub, u zasebnom repozitorijumu. Rešenje za pregledanje potrebno je da se nalaze na **main/master** grani. Repozitorijum treba da sadrži:

- Izvorni kod
- Dokumentaciju u Readme.md fajlu:
 - o Listu korišćenih tehnologija i njihovih verzija
 - o Instukcije za podešavanje okruženja potrebnog za build
 - o Instrukcije za pokretanje build-a
 - o Instrukcije za pokretanje aplikacije
 - o Instrukcije za pokretanje unit testova
- Na repozitorijum je potrebno dodati sledeće osobe kao kolaboratore sa admin rolom (podešavanje se vrši preko Settings / Collaborators and Teams / Add people):
 - o acvi9
 - o milosradojcin
 - o davorrandjelovic
 - o pavle-lukic-levi9-com
 - o ljturanjanin
 - o masadukic
 - o filipstev
 - o IvanPartalo
 - o miloradradovic-levi9
 - o oliveragavrilovic
 - o stufegdzic

Ocenjivanje i bodovanje

Prilikom ocenjivanja rešenja, biće uzeti u obzir i funkcionalni i tehnički zahtevi.

Preduslov za pregledanje zadatka i bodovanje je prisustvo kompletnog Readme file-a koji sadrži sve stavke definisane u sekciji Tehnički zahtevi.

Funkcionalni zahtevi – max 60 bodova

- Javni test scenariji (dostupan u sklopu ovog zadatka) – max 20 bodova
- Ostali test scenariji – max 40 bodova

Tehnički zahtevi – max 40 bodova

- Kvalitet rešenja i koda – max 35 bodova
- Unit testovi – max 5 bodova

Bonus poeni: Prvih deset poslatih funkcionalnih rešenja dobijaju dodatnih 5 bodova.

Slanje rešenja na ocenjivanje

Link do rešenja na GitHub-u potrebno je poslati na email: pitanja@5danauoblacima.com , subject “SPREMNO ZA PREGLEDANJE” onda kada je rešenje spremno za pregledanje. **Sve izmene nakon ovog email-a neće biti uzete u obzir. Slanje email-a je preduslov za pregledanje rešenja.** Ukoliko kandidati žele da menjaju i nadograđuju kod nakon slanja ovog email-a potrebno je da to rade na drugoj grani koja nije **main/master**.

Javni test scenariji

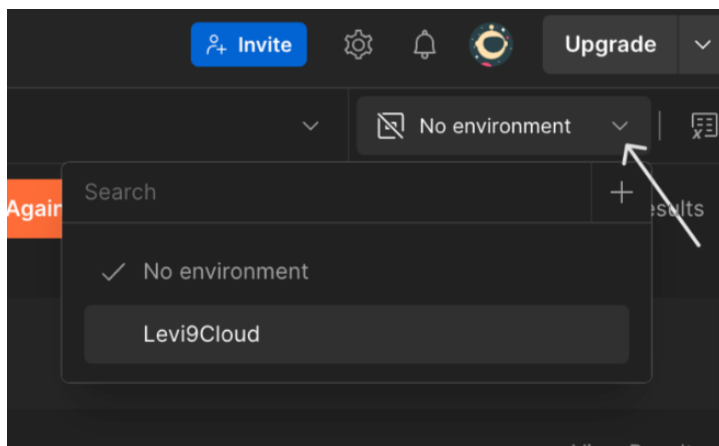
Pored ovog dokumenta svako od vas će dobiti zip u kojem se nalaze Postman kolekcija i environments fajl koji su potrebni za pokretanje javnih test scenarija.

URL za skidanje Postmana: <https://www.postman.com>

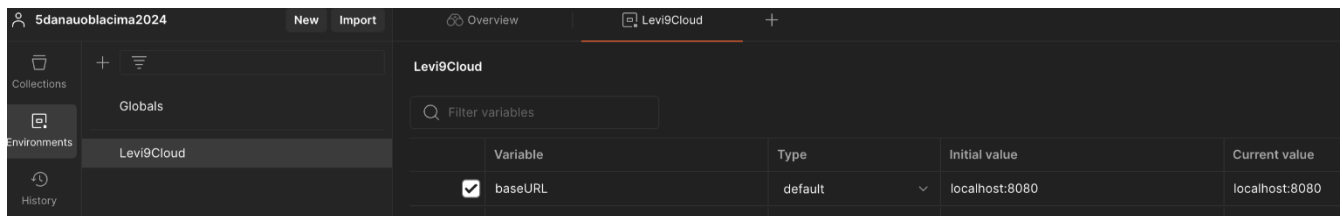
Nakon pokretanja Postmana uvucite kolekciju i environment fajlove. File -> Import i izaberite oba fajla sa .json ekstenzijom.

Nakon importa proverite da li vam se u Collections i Environments delu nalaze importovana kolekcija i environment varijabla.

Nakon toga izaberite importovani environment klikom na sledeće dugme i izaberite Levi9Cloud:

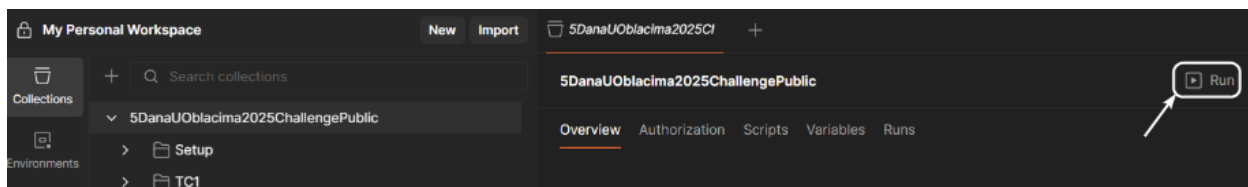


Ukoliko vam aplikacija **ne** trči na portu 8080, promenite baseURL varijablu u Environments.

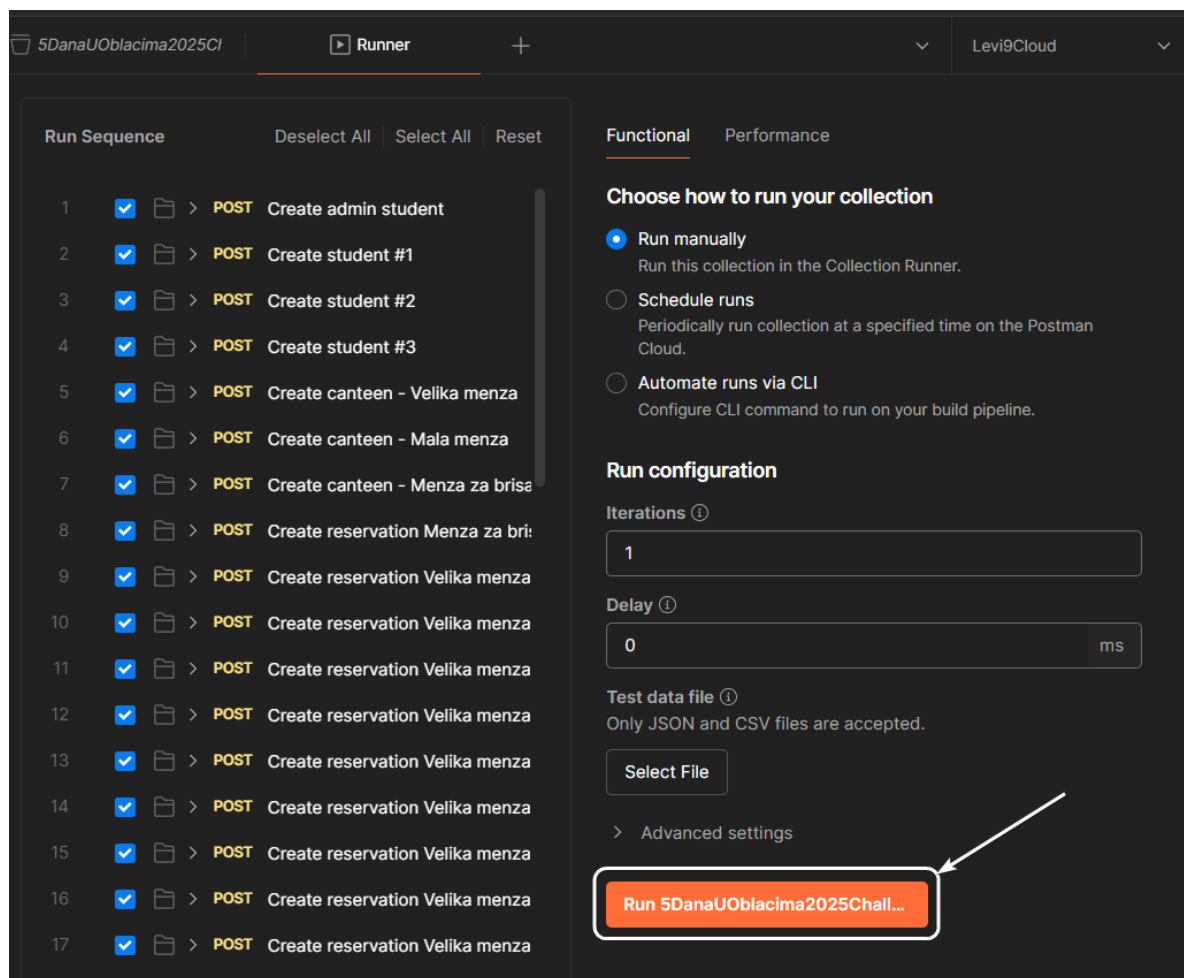


U trenutku pokretanja testova baza mora biti prazna. Ovo važi kako za prvo pokretanje testova tako i za svako ponovno pokretanje tj. re-runovanje. Potrebno je isprazniti bazu svaki put.

Da bi ste pokrenuli testove u Collections kliknite na kolekciju „**5DanaUOblacima2025ChallengePublic**” i pritisnite dugme **Run**.



Nakon toga kada se otvori prompt za pokretanje testova izaberite opciju „**Run manually**“ i kliknite na dugme „**Run 5DanaUOblacima2025ChallengePublic**”.



Testovi koji su prošli biće obeleženi zelenom bojom, a testovi koji su pali crvenom.

Jako je bitno da ne menjate redosled testova i foldera unutar kolekcije, kako u Collections ekranu tako i u Runner ekranu na slici iznad. Ukoliko slučajno to uradite a niste sigurni koji je redosled bio, obrišite kolekciju pa ponovo uradite import.

Obratite pažnju da svi testovi iz foldera *Setup* (koji se prvi izvršavaju) budu uspešni (prolaze), jer je to glavni preduslov da i ostali testovi prođu. Ukoliko neki test u *Setup*-u pada, prvo rešite taj problem, pa tek onda proveravajte rezultate ostalih scenarija.