

Sa FTP servera (folder **Download/RSII**, user/pass: **student_dm**) preuzmite template projekta na kome ćete raditi implementaciju funkcionalnosti opisanih u nastavku. Promijenite konekcijski string kako biste se povezali na SQL server na kojem ćete razvijati i testirati svoj projekt.

- **Adresa i naziv servera:** IP_ADRESA_SERVERA\ispiti
- **Port:** BROJ_PORTA
- **Baza podataka:** Koristeći migracije kreirajte bazu podataka sa vasim brojem indeksa npr. IB150051
- **Pristupni podaci:**
Username = rsII
Password = Ispit2025

Svi studenti ispit rade na **istoj instanci servera**, pa je važno da radite isključivo unutar baze koja pripada vašem projektu.

1. Dodajte inicijalnu migraciju i kreirajte vašu bazu (*update-database*). Nakon uspješne migracije trebalo bi da postoje pristupni podaci *username=admin password=Test123*
2. Trenutna verzija projekta posjeduje entitete (proizvod, korisnik, korpa) i implementaciju upravljanja korpom na strani klijenta, ali na API strani nije implementirana kompletna logika. Dovršite implementaciju logike kreiranja korpe i upravljanja njenim stavkama.
3. Kreirajte entitet **CartEventBrojIndeksa**, koji čuva podatke o svim promjenama vezanim za korpu. Entitet treba sadržavati vezu na korpu, vezu na stavku korpe (ako postoji), vezu na korisnika, tip događaja (dodavanje proizvoda, uklanjanje proizvoda, promjena količine, čišćenje korpe, checkout), vrijeme nastanka događaja, osnovne podatke o proizvodu u tom trenutku (naziv, cijena), te prethodnu i novu količinu (ukoliko je došlo do promjene).
4. Osigurati da se pri svakoj promjeni u korpi (dodavanje proizvoda, uklanjanje...) automatski kreira novi zapis u **CartEventBrojIndeksa**.
5. U okviru Flutter projekta (UI/ecommerce_mobile) kreirati formu **frmCartEventsBrojIndeksa** za pregled svih događaja nad korpama. Tom prilikom prikazati: korisnika (ime i prezime), korpu/proizvod, tip događaja, vrijeme događaja. Omogućiti filtriranje po tipu događaja i po korisniku.
6. Na formi *Shopping Cart*, u headre sekciji, dodati dugmić *Clear cart* koji će očistiti sadržaj korpe tj. ukloniti sve dodate proizvode. Tom prilikom evidentirati događaj za svaki od uklonjenih proizvoda.
7. Dugmić *Proceed to Checkout*, koji već postoji na formi, će samo evidentirati događaj za tu korpu i samim tim onemogućiti promjene na stanju korpe. Drugim riječima, osim evidencije događaja, nije potrebno implementirati bilo šta drugo od Checkout poslovne logike.

8. Dodajte par testnih podataka (seed) za novokreirane entitete.

9. Na kraju ispita, u API projektu, obrišite bin i obj foldere (opcija Clean Solution) u okviru svakog projekta (prikazan folder Model, ali uraditi i za Services i API), a unutar Flutter projekta izvršite komandu „flutter clean“.

Nakon toga, zapakujte projekat u folder imenovan vašim brojem indeksa i postavite na FTP server (folder **Upload/RSII**, user/pass: **student_dm**)

