Es-dnevnik analiza zahteva

Aplikacija treba da omogući pristup za dva tipa korisnika. Sa jedne strane, aplikaciji pristupaju **nastavnici**, a sa druge **učenici**. I nastavnici i učenici se na veb portal prijavljuju kroz istu login stranicu, koja na osnovu **korisničkog imena** utvrdi da li je reč o nastavniku ili učeniku i prema tome proverava ispravnost pristupnih parametara i kreira sesiju. Nastavnici mogu da upisuju **ocene** učenicima samo za predmete koji su im dodeljeni i to samo za učenike kojima je u bazi podataka evidentirano da prate predmet tog nastavnika. *Baza podataka predmeta, učenika i nastavnika treba da bude unapred pripremljena i popunjena, kao i veze između njih, jer ovaj projektni zadatak čini samo jedan modul većeg informacionog sistema i nije u njegovoj nadležnosti da uređuje spiskove učenika, nastavnika predmeta i veza koje uređuju koji učenik prati koji predmet, kao i toga koji nastavnik je zadužen za koji predmet*. Potrebno je da se podaci o ocenama evidentiraju trajno, bez mogućnosti brisanja ili prepravljanja i da se vrši višestruko insistiranje na potvrdi namere unosa konkretne ocene koju nastavnik upisuje učeniku za konkretan predmet. Aplikacija treba da automatski računa ocenu učenika na osnovu svih njegovih ocena i to za svako tromesečje, kao i na osnovu tih ocena da predlaže ocenu koja treba da bude zaključena u dnevnik. Nastavnik može da odluči da prihvati predloženu ocenu ili da uz posebno **obrazloženje** (koje se evidentira u bazu) odluči učeniku da unese ocenu koja je veća od predložene. Učenik kada se prijavi na portal može da vidi spisak svih svojih ocena, podeljenih po tromesečjima za svaki od predmeta, kao i **zaključenu ocenu** za svaki od predmeta ako takva postoji. Grafički interfejs veb sajta treba da bude realizovan sa responsive dizajnom.  
  
  
Tehnička ograničenja  
- Aplikacija mora da bude realizovana na Node.js platformi korišćenjem Express biblioteke. Aplikacija mora da bude podeljena u dve nezavisne celine: back-end veb servis (API) i front-end (GUI aplikacija). Sav kôd aplikacije treba da bude organizovan u jednom Git spremištu u okviru korisničkog naloga za ovaj projekat, sa podelom kao u primeru zadatka sa vežbi.  
- Baza podataka mora da bude relaciona i treba koristiti MySQL ili MariaDB sistem za upravljanje bazama podataka (RDBMS) i u spremištu back-end dela aplikacije mora da bude dostupan SQL dump strukture baze podataka, eventualno sa inicijalnim podacima, potrebnim za demonstraciju rada projekta.  
- Back-end i front-end delovi projekta moraju da budi pisani na TypeScript jeziku, prevedeni TypeScript prevodiocem na adekvatan JavaScript. Back-end deo aplikacije, preveden na JavaScript iz izvornog TypeScript koda se pokreće kao Node.js aplikacija, a front-end deo se statički servira sa rute statičkih resursa back-end dela aplikacije i izvršava se na strani klijenta. Za postupak provere identiteta korisnika koji upućuje zahteve back-end delu aplikacije može da se koristi mehanizam sesija ili JWT (JSON Web Tokena), po slobodnom izboru.  
- Sav generisani HTML kôd koji proizvodi front-end deo aplikacije mora da bude 100% validan, tj. da prođe proveru W3C Validatorom (dopuštena su upozorenja - Warning, ali ne i greške - Error). Grafički korisnički interfejs se generiše na strani klijenta (client side rendering), korišćenjem React biblioteke, dok podatke doprema asinhrono iz back-end dela aplikacije (iz API-ja). Nije neophodno baviti se izradom posebnog dizajna grafičkog interfejsa aplikacije, već je moguće koristiti CSS biblioteke kao što je Bootstrap CSS biblioteka. Front-end deo aplikacije treba da bude realizovan tako da se prilagođava različitim veličinama ekrana (responsive design).  
- Potrebno je obezbediti proveru podataka koji se od korisnika iz front-end dela upućuju back-end delu aplikacije. Moguća su tri sloja zaštite i to: (1) JavaScript validacija vrednosti na front-end-u; (2) Provera korišćenjem adekvatnih testova ili regularnih izraza na strani servera u back-end-u (moguće je i korišćenjem izričitih šema - Schema za validaciju ili drugim pristupima) i (3) provera na nivou baze podataka korišćenjem okidača nad samim tabelama baze podataka.  
- Neophodno je napisati prateću projektnu dokumentaciju o izradi aplikacije koja sadrži (1) model baze podataka sa detaljnim opisom svih tabela, njihovih polja i relacija; (2) dijagram baze podataka; (3) dijagram organizacije delova sistema, gde se vidi veza između baze, back-end, front-end i korisnika sa opisom smera kretanja informacija; (4) popis svih aktivnosti koje su podržane kroz aplikaciju za sve uloge korisnika aplikacije prikazane u obliku Use-Case dijagrama; kao i (5) sve ostale elemente dokumentacije predviđene uputstvom za izradu dokumentacije po ISO standardu.  
- Izrada oba dela aplikacije (projekata) i promene kodova datoteka tih projekata moraju da bude praćene korišćenjem alata za verziranje koda Git, a kompletan kôd aplikacije bude dostupan na javnom Git spremištu, npr. na besplatnim GitHub ili Bitbucket servisima, jedno spremište za back-end projekat i jedno za front-end projekat. Ne može ceo projekat da bude otpremljen u samo nekoliko masovnih Git commit-a, već mora da bude pokazano da je projekat realizovan u kontinuitetu, da su korišćene grane (branching), da je bilo paralelnog rada u više grana koje su spojene (merging) sa ili bez konflikata (conflict resolution).

Uloge: posetilac

Loguje se

nastavnik

Upisuje ocene

Zaključuje ocene

Dodaje beleške

Pregleda ocene

Učenik

Pregleda ocene

Entiteti:

Predmet subject subject\_id\_pk

Naziv name

Nastavnik professor

Ime, prezime, predmet, username, password professor\_id\_pk

name

subject\_id\_fk

surname

username\_uq

hash\_password

Učenik student

Ime, prezime, username, password student\_id\_pk

name

surname

username\_uq

hash\_password

Učenik-predmet: student\_subject

student\_subject\_id\_pk

ocene grade

subject\_id\_fk\_uq1

student\_id\_fk\_uq1

Nastavnik-učenik-predmet: professor\_student\_subject

professor\_student\_subject\_id\_pk

zaključna ocena final\_grade

beleška note

professor\_id\_fk\_uq1  
student\_subject\_id\_fk\_uq1

Profesor-učenik professor\_student

professor\_student\_id\_pk

professor\_id\_fk\_uq1

student\_id\_fk\_uq1

Ograničenja:

professor\_student\_subject

professor\_student

professor

N:N

subject

N:N

student\_subject

student

1:N

posetilac

profesor

učenik