perplexity

RAF Event Booker - Analiza Projekta i Plan Realizacije

Na osnovu detaljne analize vašeg projektnog zadatka, evo sveobuhvatnog plana realizacije sa logičkim redosledom komponenti i potencijalnim problemima:

1. OSNOVNA ANALIZA PROJEKTA

Vaš projekat se sastoji od **dva dela**:

- **EMS (Event Management System)** - backend za autentifikovane korisnike
- **Javna platforma** - frontend dostupan svima

2. TEHNOLOGIJE I VEŠTINE KOJE MORATE SAVLADATI

Backend tehnologije (izaberite jednu):

- **JAX-RS** - Java REST framework
- **Node.js + Express** - JavaScript runtime + web framework
- **SparkJava** - minimalni Java web framework

Frontend tehnologije (izaberite jednu):

- **Vue.js** - progresivni JavaScript framework
- **React.js** - JavaScript biblioteka za UI
- **jQuery** - jednostavna JavaScript biblioteka
- **Angular** - kompletan MVC framework

Dodatne veštine:

- **Relaciona baza podataka** (MySQL/PostgreSQL)
- **Normalizacija baze** (do 3NF)
- **HTTP/REST protokoli**
- **Autentifikacija i autorizacija**
- **Paginacija**
- **Like/dislike sistemi**
- **RSVP funkcionalnost**

3. LOGIČAN REDOSLED IZRADE KOMPONENTI

FAZA 1: Priprema i setup (Nedelja 1-2)

1.1. Analiza zahteva i dizajn baze

```
Entiteti:
- korisnik (id, email, ime, prezime, tip, status, lozinka_hash)
- kategorija (id, naziv, opis)
- događaj (id, naslov, opis, datum_kreiranja, datum_održavanja, lokacija, views, autor_id
- tag (id, naziv)
- događaj_tag (događaj_id, tag_id)
- komentar (id, ime_autora, tekst, datum_kreiranja, događaj_id, like_count, dislike_count
- rsvp (id, korisnički_identifikator, događaj_id, datum_prijave)
- reakcija (id, tip_objekta, objekat_id, tip_reakcije, identifikator_korisnika)
```

Potencijalni problemi:

- Nedoslednost u normalizaciji[1] [2]
- Pogrešno modelovanje many-to-many veza
- Neoptimalne foreign key constrainte[3]

1.2. Setup razvojnog okruženja

- Instalacija odabrane backend tehnologije
- Setup baze podataka
- Kreiranje osnovne folder strukture[4] [5]

Potencijalni problemi:

- Verzijski konflikti dependency-ja
- Konfiguracija development vs production okruženja
- Port konflikti između servisa

FAZA 2: Backend osnove (Nedelja 3-4)

2.1. Kreiranje osnovnih REST endpoints-a

```
// Primer strukture za Node.js/Express
/api/auth/login
/api/auth/logout
/api/categories
/api/events
/api/users (samo admin)
```

2.2. Implementacija autentifikacije[6] [7]

- Kreiranje login/logout funkcionalnosti

- Session ili JWT token management
- Authorization middleware

Potencijalni problemi:

- Nepravilno hashovanje lozinki
- Session expiry ne radi pravilno[8]
- Authorization bypass vulnerabilities
- CORS problemi između frontend-a i backend-a

2.3. CRUD operacije za kategorije

- Kreiranje, čitanje, ažuriranje, brisanje kategorija
- Validacija da kategorija ne može biti obrisana ako ima događaje

Potencijalni problemi:

- Race conditions pri brisanju
- Nedosledna validacija između frontend-a i backend-a[9]
- SQL injection mogućnosti

FAZA 3: Core funkcionalnost (Nedelja 5-7)

3.1. Upravljanje događajima

- CRUD operacije za događaje
- Tag system implementacija
- Upload i handling tagova

Potencijalni problemi:

- Tag parsing iz comma-separated stringova
- Dupliciranje tagova
- Performance problemi sa velikim brojem tagova

3.2. Search funkcionalnost

- Pretraga po naslovu i opisu
- Paginacija rezultata[10] [11] [12]

Potencijalni problemi:

- Spore SQL queries bez proper indexa
- Paginacija ne radi sa search filterima
- Memory issues sa velikim result setovima

3.3. Upravljanje korisnicima (Admin)

- CRUD operacije za korisnike

- Aktivacija/deaktivacija korisnika

Potencijalni problemi:

- Admin ne može da deaktivira sebe
- Password validation nedoslednost
- Email uniqueness constraint problemi

FAZA 4: Javna platforma (Nedelja 8-10)

4.1. Osnovni prikaz događaja

- Home page sa najnovijim događajima
- Kategorizovani prikaz
- Search funkcionalnost

4.2. View counting sistem [13] [14] [15]

- Cookie/session based tracking
- Jedinstveni view per visitor

Potencijalni problemi:

- Cookie blokiranje od strane browsera [13]
- Bot traffic inflatira view counts
- Session handling problemi u distributed setup

4.3. Like/Dislike sistem [16] [17] [18] [19]

- Database design za reactions
- Frontend AJAX implementacija
- Dupliciranje prevencija

Potencijalni problemi:

- Race conditions kod simultanih like/dislike akcija
- Client-side manipulation (potrebna server validacija)
- Database design za skalabilnost [19]

FAZA 5: Napredne funkcionalnosti (Nedelja 11-12)

5.1. Komentari sistem

- CRUD za komentare
- Like/dislike na komentare
- Validacija sadržaja

Potencijalni problemi:

- XSS napadi kroz komentare
- Spam prevention
- Comment threading (ako se dodaje)

5.2. RSVP sistem[20] [21] [22]

- Database design za RSVP
- Kapacitet management
- Email/notification sistem

Potencijalni problemi:

- Overbooking kada je max_kapacitet definisan
- Race conditions pri RSVP prijavi
- Email delivery failures
- GDPR compliance za čuvanje email adresa

5.3. "Pročitaj još" i related events

- Algoritam za pronalaženje sličnih događaja
- Performance optimization

Potencijalni problemi:

- N+1 query problemi
- Algoritam za similarity može biti spor
- Cache invalidation

FAZA 6: Finalizacija i deploy (Nedelja 13-14)

6.1. Error handling i validacija[23] [9] [24]

- Centralizovano error handling
- Input validacija
- Proper HTTP status codes

6.2. Database migrations i schema updates[25] [26] [27]

- Migration scripts
- Data consistency checks
- Backup strategije

6.3. Testing i performance optimization

- Unit testovi za kritične komponente
- Load testing
- Database query optimization

4. NAJVEROJATNIJI PROBLEMI PO KOMPONENTAMA

Autentifikacija i autorizacija:

- Session hijacking vulnerabilities
- Password hashing inconsistencies
- JWT token expiry handling[6]
- Authorization middleware bypass

Database design i performance:

- Normalization problemi[1] [28] [29]
- Spore queries zbog nedostajućih indeksa
- Foreign key constraint violations[3]
- Migration script failures[25] [27]

Like/Dislike sistem:

- Race conditions pri simultanim akcijama[16] [19]
- Database struktura ne skalira[19]
- Client-side manipulation
- Cookie/session management problemi

RSVP funkcionalnost:

- Overbooking scenariji[20] [22]
- Email delivery failures
- Data privacy compliance
- Concurrent booking conflicts

Search i paginacija:

- Performance degradacija sa velikim datasetovima[12]
- Inconsistent pagination sa filterima[10] [11]
- Search relevance problemi
- Memory usage sa nested queries

View tracking:

- Bot traffic inflating numbers[13] [14]
- Cookie consent law compliance[15]
- Session tracking across devices

- Database growth problemi

Error handling:

- Information leakage kroz error messages[23] [9]
- Inconsistent validation između frontend-a i backend-a
- Unhandled edge cases
- Poor user experience sa generic error messages

General development problemi:

- Folder structure postaje neorganizovana[4] [30] [31]
- Frontend/backend API contract inconsistencies
- Environment configuration problemi
- Dependency version conflicts

5. PREPORUKE ZA USPEŠNU REALIZACIJU

1. **Počnite sa jednostavnim MVP-om** - implementirajte osnovne CRUD operacije pre naprednih funkcionalnosti
2. **Testirajte svaku komponentu pre prelaska na sledeću**
3. **Dokumentujte APIs dok ih pravite**
4. **Koristite version control (Git) od početka**
5. **Implementirajte error handling rano** - ne ostavljajte za kraj
6. **Pravite frequent backups baze podataka**
7. **Testirajte edge cases** - šta se dešava kada nema podataka, kad je baza nedostupna, itd.

Ovaj plan vam omogućava iterativni razvoj gde možete testirati svaku komponentu pre prelaska na sledeću, što značajno smanjuje rizik od velikih problema kasno u projektu.

⁂

1. http://www.mecs-press.org/ijem/ijem-v7-n5/v7n5-6.html
2. https://dl.acm.org/doi/10.1145/3744897
3. https://www.bytebase.com/blog/top-database-schema-design-best-practices/
4. https://www.geeksforgeeks.org/javascript/file-and-folder-organization-best-practices-for-web-development/
5. https://nextjs.org/docs/app/getting-started/project-structure
6. https://dev.to/codeparrot/jwt-vs-session-authentication-1mol
7. https://clerk.com/blog/combining-the-benefits-of-session-tokens-and-jwts
8. https://www.reddit.com/r/node/comments/1aox0au/whats_the_ultimate_resource_for_jwt_vs_session/
9. https://treblle.com/blog/rest-api-error-handling

10. https://www.juniper.net/documentation/us/en/software/mist/automation-integration/topics/concept/rest-api-pagination.html
11. https://www.baeldung.com/rest-api-pagination-in-spring
12. https://apisyouwonthate.com/blog/api-design-basics-pagination/
13. https://community.hubspot.com/t5/Reporting-Analytics/Will-HubSpot-track-sessions-if-users-decline-or-ignore-the/m-p/1069619
14. https://stackoverflow.com/questions/9355444/keep-track-of-user-page-view-using-cookie-php
15. https://breakdance.com/documentation/advanced/disable-breakdance-cookies/
16. https://codewithawa.com/posts/like-dislike-system-with-php-and--mysql
17. https://www.itsolutionstuff.com/post/laravel-12-like-dislike-system-tutorial-exampleexample.html
18. https://www.youtube.com/watch?v=OILXsP8Hrek
19. https://stackoverflow.com/questions/14202168/sql-database-structure-for-like-and-dislike
20. https://www.socioplace.com/events
21. https://godreamcast.com/blog/solution/in-person-event/event-rsvp-management-tips/
22. https://www.gevme.com/en/blog/automated-vs-manual-event-rsvp-services-which-is-right-for-your-organization/
23. https://www.ibm.com/docs/en/app-connect/12.0.x?topic=apis-implementing-error-handler-in-rest-api
24. https://www.baeldung.com/rest-api-error-handling-best-practices
25. https://www.prisma.io/dataguide/types/relational/migration-strategies
26. https://supabase.com/docs/guides/deployment/database-migrations
27. https://learn.microsoft.com/en-us/ef/core/managing-schemas/migrations/
28. https://en.wikipedia.org/wiki/Database_normalization
29. https://www.geeksforgeeks.org/dbms/introduction-of-database-normalization/
30. https://www.reddit.com/r/webdev/comments/1kg5hc7/best_practices_for_organizing_large_web_projects/
31. https://saipranay47.hashnode.dev/web-app-folder-structure-a-modern-approach