EVENT ORGANISER APP

RAPORT

ILIE IOAN-CALIN 333AA

**Aplicația Events Organiser** este o platformă web construită pe **arhitectura MVC (Model-View-Controller)**, dedicată organizării și gestionării evenimentelor. Dezvoltată folosind **Java Spring Boot** pentru backend și **Thymeleaf** pentru interfața, aplicația permite utilizatorilor să creeze, să editeze, să vizualizeze și să gestioneze evenimente într-un mod eficient și intuitiv. Este destinată persoanelor care organizează evenimente de tip petreceri, ședințe, întâlniri business și altele.

1. **Pachete și clase:**

* **Clasa principală:**
  + *EventsOrganiserApplication*: Punctul de intrare al aplicației, marchează aplicația ca Spring Boot Application. Conține metoda main care inițializează contextul Spring.
* **Pachetul repository:**
  + *EventRepository*: Interfață pentru accesul la datele evenimentelor. Extinde JpaRepository și oferă metode de interogare pentru entitatea Event.
  + *EventExtrasRepository*: Interfață pentru gestionarea opțiunilor suplimentare ale evenimentelor. Extinde JpaRepository.
  + *UserRepository*: Interfață pentru accesul la datele utilizatorilor. Conține metode pentru autentificare și gestionarea profilurilor.
* **Pachetul model:**
  + *Event*: Definește structura unui eveniment. Atribute principale:
    - id: ID-ul unic al evenimentului.
    - eventName: Numele evenimentului.
    - eventDescription: Descrierea evenimentului.
    - eventType: Tipul evenimentului (party, bussines meeting, workshop, conference).
    - eventDate: Data evenimentului.
    - idUser: ID-ul utilizatorului care organizează evenimentul.
  + *EventExtras*: Reprezintă opțiunile suplimentare selectate pentru evenimente (speakers, drones, lighting, catering).
  + *User*: Reprezintă utilizatorii aplicației. Atribute principale:
    - Id: Id-ul unic al utilizatorului.
    - username: Numele utilizatorului
    - email: Adresa de email.
    - password: parola utilizatorului
  + *LoginRequest*: Model pentru autentificare. Conține câmpuri precum email și parolă.
* **Pachetul service:**
  + *UserService*: Gestionează logica prntru profilele de utilizatori. Funcționalități:
    - Înregistrarea unui nou utilizator.
    - Autentificare.
    - Gestionarea profilului utilizatorului.
  + *EventService:* Gestionează evenimentele. Funcționalități:
    - Adăugarea unui nou eveniment.
    - Vizualizarea, editarea, și ștergerea evenimentelor existente.
* **Pachetul controller:**
  + *LoginController*: Controlează procesul de autentificare a utilizatorilor.
  + *UserController*: Oferă funcționalități pentru gestionarea conturilor utilizatorilor.
  + *HomeController:* Gestionează afișarea paginii principale.
  + *DashboardController:* Controlează vizualizarea evenimentelor organizate de utilizator.
  + *NewEventController:* Permite utilizatorului să creeze evenimente noi.
  + *EventController*: Controlează operațiunile CRUD (Creare, Citire, Actualizare, Ștergere) pentru evenimente.
  + *ProfileController*: Permite gestionarea profilului utilizatorului logat.
  + *LogoutController*: Oferă funcționalitatea de deconectare.
* **Pachetul util:**
  + *UserValidator*:  Clasă pentru validarea datelor introduse de utilizator (formatul parolei, adresa de email validă etc.).
* **Pachetul config:**
  + SecurityConfig: Configurează securitatea aplicației. Include setările pentru autentificare și autorizare.

1. **Fișiere HTML și CSS (static folder)**
   1. *index.html*: Pagina principală a aplicației. Prezintă o introducere și opțiuni de navigare pentru utilizatori.
   2. *profile.html*: Pagina de profil a utilizatorului. Permite vizualizarea și actualizarea detaliilor personale.
   3. *new\_event.html*: Pagina de creare a unui eveniment. Include formulare pentru detaliile evenimentului și opțiuni suplimentare.
   4. *dashboard.html*: Dashboard-ul utilizatorului. Listează toate evenimentele organizate de acesta și oferă opțiuni de editare/ștergere.
   5. *payment\_page.html*: Pagina pentru procesarea plăților aferente evenimentelor.
   6. *edit\_event.html*: Pagina pentru editarea detaliilor unui eveniment existent.
   7. *loginpage.html*: Pagina de autentificare. Permite utilizatorilor să se conecteze în aplicație.
   8. *confirm\_delete.html*: Afișează un dialog de confirmare pentru ștergerea unui eveniment.
   9. *signup.html*: Pagina de înregistrare a unui nou utilizator. Include validări pentru datele introduse.

**Funcționalități oferite prin interfața grafică:**

1. Autentificare și înregistrare:
   1. Pagina de login permite autentificarea.
   2. Pagina de signup oferă validări pentru datele utilizatorilor la înregistrarea pe site.
2. Crearea și gestionarea evenimentelor:
   1. Formular pentru adăugarea unui nou eveniment.
   2. Dashboard pentru vizualizarea evenimentelor organizate.
   3. Editarea și ștergerea evenimentelor.
3. Procesare plăți:
   1. Pagina de plată oferă o interfață pentru selecția opțiunilor suplimentare.
4. Gestionarea profilului:
   1. Permite utilizatorului să își actualizeze datele personale.
5. Confirmări și dialoguri:
   1. Dialoguri de confirmare pentru operații critice, precum ștergerea unui eveniment.
6. **Testarea aplicației**
   1. **Testare automată:**
      * *EventsOrganiserApplicationTests*: Teste de bază pentru verificarea inițializării aplicației și contextului Spring.
      * *UserValidatorTest*: Verifică validarea datelor utilizatorilor, cum ar fi:
        + Formatul parolei.
        + Formatul adresei de email.

* *ProfileControllerTest*: Teste pentru metodele din controllerul profilului, incluzând:
  + Actualizarea datelor de profil.
  + Accesarea detaliilor profilului.
* *UserServiceTest*: Verifică metodele serviciului pentru utilizatori:
  + Înregistrare.
  + Autentificare.
  + Gestionarea datelor utilizatorilor.
* **Testare manuală:**
  + Autentificare și înregistrare:
    - Verificarea scenariilor de succes și eșec:
    - Date valide → acces permis.
    - Date invalide (parolă scurtă, email greșit) → eroare afișată.
  + Crearea unui eveniment:
    - Crearea unui eveniment cu detalii complete.
    - Lăsarea unor câmpuri goale → afișarea mesajelor de eroare.
  + Editarea și ștergerea unui eveniment:
    - Testarea actualizării detaliilor unui eveniment.
    - Confirmarea operației de ștergere.
  + Procesare plăți:
    - Testarea afișării opțiunilor suplimentare selectate și navigarea spre pagina de plată.
  + Interfața grafică:
    - Verificarea navigării între pagini.
    - Responsivitatea și afișarea corectă a elementelor vizuale.
  + Erori gestionate:
    - Acces la pagini restricționate fără autentificare → redirecționare către login.

1. **Idei pentru  îmbunătățirea aplicației**
   1. Implementarea caching-ului:
      * Folosirea unui tool pentru a reduce interogările frecvente la baza de date și a îmbunătăți viteza aplicației.
   2. Pagini încărcate asincron:
      * Reducerea timpului de încărcare al paginilor prin utilizarea cererilor asincrone pentru actualizarea conținutului, în special în dashboard.
   3. Politici stricte de gestionare a sesiunilor:
      * Utilizarea token-urilor JWT pentru autentificare și implementarea unei expirări automate pentru a spori securitatea.
   4. Testare automată extinsă:
      * Crearea mai multor teste automate pentru cazuri edge, inclusiv pentru gestionarea erorilor și securitatea aplicației.
   5. Funcționalitate de partajare a evenimentelor**:** 
      * Adăugarea opțiunii de a partaja evenimentele pe rețele sociale (e.g., Facebook, Twitter) sau printr-un link unic generat de aplicație.