**PLATFORMĂ INTEGRATĂ PENTRU GESTIONAREA ŞI VIZUALIZAREA CURSURILOR VALUTARE**

Software Design Document

|  |  |  |
| --- | --- | --- |
|  | Contribuitori: | **Bogdan-Alexandru BÂRGĂOANU**  **Diana-Susana FORRAI** |
|  |  |  |
|  | Coordonator științific: | **Prof. dr. ing. Ioan-Valentin SITA** |

**Cluj-Napoca, 2024**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | |  | |
|  |  | |  | |
|  |  |  | |

**PLATFORMĂ INTEGRATĂ PENTRU GESTIONAREA ŞI VIZUALIZAREA CURSURILOR VALUTARE**

1. **Enunțul temei:** *Dezvoltarea unei aplicații web full stack care să permită monitorizarea în timp real și gestionarea cursurilor de schimb valutar, incluzând un API pentru preluarea și stocarea datelor, o interfață client pentru utilizatori finali și un panou de administrare pentru gestionarea resurselor și configurărilor sistemului.*
2. **Conținutul lucrării:** *Pagina de prezentare, introducere, obiectivele lucrării, studiu bibliografic, analiză teoretică și sinteză,* *Proiectare de detaliu și implementare,* *manual de instalare și utilizare, concluzii, bibliografie, anexe.*
3. **Locul documentării**: Universitatea Tehnică din Cluj-Napoca
4. **Data emiterii temei:** 30 septembrie 2024
5. **Data predării:** 20 decembrie 2024

Cuprins

[Capitolul 1. Introducere (Scopul Proiectului) 1](#_Toc186734761)

[Capitolul 2. Obiectivele proiectului 2](#_Toc186734762)

[Capitolul 3. Conținutul Proiectului 4](#_Toc186734763)

[Capitolul 4. Modelul Datelor 6](#_Toc186734764)

[4.1. Aplicația Server / API 6](#_Toc186734765)

[4.1.1. Structura bazei de date 6](#_Toc186734766)

[4.1.2. Relații între tabele: 6](#_Toc186734767)

[4.1.3. Securitatea datelor 7](#_Toc186734768)

[Capitolul 5. Modelul Arhitectural 8](#_Toc186734769)

[5.1. Plan de proiectare și implementare 8](#_Toc186734770)

[5.2. Aplicația Server / API 9](#_Toc186734771)

[5.3. Aplicația Administrator 11](#_Toc186734772)

[5.4. Aplicația Client 13](#_Toc186734773)

[Capitolul 6. Modelul interfetei cu utilizatorul 16](#_Toc186734774)

[6.1. Aplicația Administrator 16](#_Toc186734775)

[6.2. Aplicatia Client 19](#_Toc186734776)

[Capitolul 7. Elementele de testare 21](#_Toc186734777)

[Anexe 23](#_Toc186734778)

# Introducere (Scopul Proiectului)

Acest proiect propune dezvoltarea unei aplicații web full-stack pentru monitorizarea și gestionarea cursurilor de schimb valutar. Soluția integrează trei componente principale: un API backend care gestionează cererile și interacțiunile cu baza de date, o interfață client destinată utilizatorilor finali pentru vizualizarea cursurilor valutare în timp real și o interfață de administrare pentru gestionarea resurselor și configurarea sistemului.

Aplicația utilizează tehnologii moderne precum *Node.js* și *Express.js* pentru backend, *React.js* pentru componentele frontend, și *MySQL* pentru stocarea datelor. Funcționalitățile sunt extinse prin utilizarea *JWT* pentru securizarea API-urilor și a autentificării, precum și prin integrarea de grafice și hărți pentru o experiență interactivă a utilizatorului.

Sistemul trebuie sa ofere un grad ridicat de configurabilitate, permițând personalizarea aplicației în funcție de nevoile fiecărei societăți utilizatoare. Administratorii pot configura locațiile în mod dinamic, asociind fiecare rată valutară unei locații specifice. Această flexibilitate permite urmărirea precisă a cursurilor valutare în diferite regiuni, asigurând o gestionare detaliată a datelor financiare.

Scopul principal al platformei este de a oferi o soluție simplificată și accesibilă pentru orice utilizator, având ca obiectiv să funcționeze ca o platformă gratuită dedicată clienților. Utilizatorii pot căuta și selecta cursurile valutare de interes, beneficiind de opțiuni de filtrare avansate. Platforma permite afișarea cursurilor valutare disponibile, fie în ansamblu, fie în funcție de proximitatea față de locația utilizatorului, utilizând coordonatele GPS pentru a identifica punctele de schimb valutar din apropiere. În plus, aplicația facilitează accesul la cele mai avantajoase rate de schimb, oferind o experiență intuitivă și eficientă.

Această soluție adresează necesitatea unei platforme centralizate pentru urmărirea cursurilor de schimb valutar, oferind atât accesibilitate, cât și o modalitate eficientă de administrare a datelor financiare.

# Obiectivele proiectului

În contextul globalizării economice și al interdependenței financiare, cunoașterea și monitorizarea cursurilor de schimb valutare reprezintă un aspect esențial pentru instituțiile financiare, companiile de comerț internațional și chiar pentru utilizatorii individuali care doresc să își gestioneze mai eficient tranzacțiile financiare. De asemenea, în era digitală, accesul rapid și ușor la aceste informații devine din ce în ce mai important, iar tehnologiile moderne oferă posibilitatea de a construi platforme care să furnizeze date actualizate, ușor accesibile și personalizabile. Astfel, tema acestui proiect se centrează pe dezvoltarea unei platforme complete, capabile să urmărească și să stocheze datele referitoare la cursurile de schimb valutar, să ofere un sistem de administrare al acestora și să permită vizualizarea acestora prin intermediul unei interfețe client intuitivă și interactivă.

**Obiectivele proiectului:**

1. **Dezvoltarea unei aplicații de tip full-stack**:
   * Crearea unui backend robust care să gestioneze API-ul de acces la datele valutare și să interacționeze eficient cu baza de date *MySQL*.
   * Crearea unui frontend intuitiv, utilizând *React.js*, care să permită utilizatorilor să vizualizeze cursurile valutare actualizate și să le selecteze în funcție de locație și de preferințele lor.
2. **Gestionarea și actualizarea dinamică a datelor valutare**
   * Permite administratorilor să adauge, modifice sau să elimine cursurile de schimb și locațiile asociate acestora printr-o aplicație de administrare construită cu *React.js*.
   * Asigurarea unui proces de actualizare în timp real a cursurilor de schimb, cu date provenite din surse externe și actualizări periodice automate.
3. **Implementarea funcționalității de geolocalizare și personalizare a experienței utilizatorului**
   * Permite utilizatorilor să vizualizeze cursurile de schimb cele mai apropiate de locația lor, utilizând funcționalitățile de geolocalizare din cadrul aplicației frontend client.
   * Oferirea posibilității de a selecta cele mai bune rate de schimb, având în vedere atât locația, cât și preferințele de monedă ale utilizatorului.
4. **Securizarea aplicației**
   * Implementarea unui sistem de autentificare și autorizare bazat pe *JWT*(JSON Web Tokens**)** pentru a proteja accesul la API-uri și pentru a asigura un nivel ridicat de securitate a datelor.
   * Protejarea parolelor utilizatorilor prin criptare folosind algoritmi de tip *MD5*.
5. **Crearea unui sistem de rapoarte și vizualizări**
   * Integrarea de grafice și hărți interactive pentru vizualizarea evoluției cursurilor de schimb pe perioade determinate, utilizând biblioteci precum *React Google Charts* și *Google Maps*.
   * Oferirea posibilității de a genera rapoarte personalizate pe baza locației și monedei alese de utilizator sau administrator.
6. **Scalabilitate și performanță**
   * Asigurarea unei arhitecturi scalabile, capabile să susțină un număr mare de utilizatori și tranzacții simultane, prin optimizarea interacțiunilor cu baza de date și a performanței aplicației frontend.
   * Utilizarea unui sistem performant de stocare a datelor pentru gestionarea și procesarea eficientă a cursurilor de schimb și a altor informații financiare.

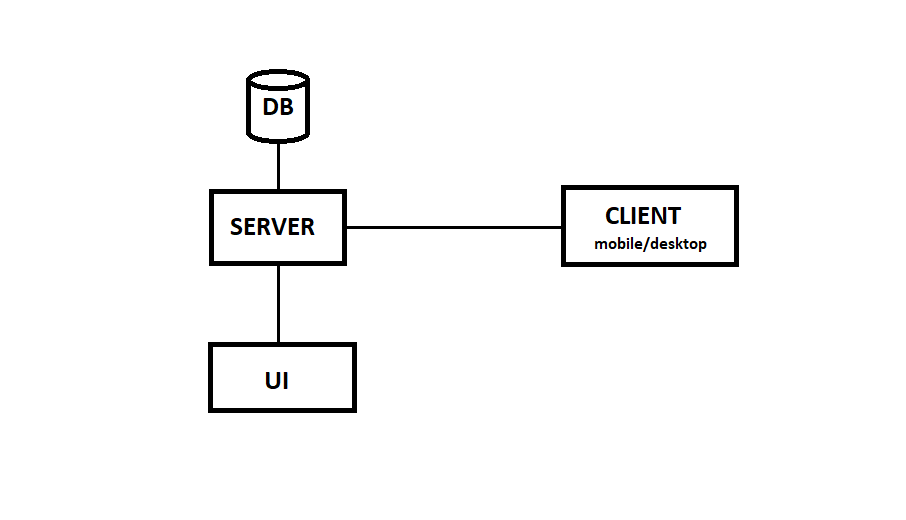


Figura 2.1. Arhitectura platformei

**Metodologia de dezvoltare:**

Proiectul va fi realizat folosind metodologia Agile, cu sprinturi de dezvoltare scurte și obiective clare. Fiecare componentă a aplicației va fi dezvoltată în paralel, urmând ca la final să fie integrate într-o soluție unitară. Testarea va fi realizată continuu, atât la nivel de unități, cât și prin teste de integrare și de performanță, pentru a asigura funcționarea corectă a întregii platforme. De asemenea, se va urmări integrarea eficientă a platformei cu surse externe de date pentru actualizarea cursurilor de schimb și a altor informații financiare relevante.

**Rezultatele așteptate:**

La finalizarea proiectului, se va obține o platformă complet funcțională, care va permite gestionarea și vizualizarea cursurilor de schimb valutare într-un mod ușor de utilizat, sigur și scalabil. Utilizatorii vor putea accesa datele în timp real, iar administratorii vor avea la dispoziție un sistem eficient pentru actualizarea și gestionarea acestora. Platforma va contribui la îmbunătățirea accesului la informații financiare relevante, oferind în același timp o soluție completă pentru monitorizarea și analiza cursurilor de schimb pe baza locației și a altor parametri relevanți.

# Conținutul Proiectului

Aplicația noastră web oferă o soluție modernă și eficientă pentru a ține pasul cu fluctuațiile constante ale pieței valutare. Concepută atât pentru utilizatorii individuali, cât și pentru companiile care efectuează tranzacții internaționale, aplicația pune la dispoziție informații actualizate în timp real despre cursurile de schimb ale unei game variate de valute.

**Arhitectură și Funcționalități**

Aplicația este construită pe o arhitectură robustă, împărțită în trei componente principale:

1. **API Backend:**
   * **Interfața cu baza de date:** Stochează și actualizează constant datele despre cursurile de schimb, provenind din surse externe de încredere.
   * **Gestionarea cererilor:** Procesează solicitările venite din partea interfeței utilizatorului și a interfeței de administrare, asigurând o comunicare rapidă și sigură.
   * **Securizare:** Implementează mecanisme de autentificare și autorizare pentru a proteja datele confidențiale.
2. **Interfața Utilizatorului:**
   * **Vizualizare în timp real:** Afișează cursurile de schimb actualizate constant pentru o gamă largă de perechi valutare.
   * **Personalizare:** Permite utilizatorilor să își configureze propriile preferințe de afișare, urmărind doar perechile valutare de interes.
   * **Alerte:** Notifică utilizatorii prin e-mail sau notificări push atunci când o anumită pereche valutară atinge un prag predefinit.
   * **Grafice și analize:** Oferă instrumente vizuale pentru a analiza tendințele și evoluția cursurilor de schimb în timp.
   * **Conversie valutară:** Permite utilizatorilor să convertească sume între diferite valute, utilizând cursurile de schimb actualizate.
   * **Localizare:** Identifică automat locația utilizatorului și afișează cursurile relevante pentru regiunea respectivă.
3. **Interfața de Administrare:**
   * **Gestionarea utilizatorilor:** Creează, modifică și șterge conturi de utilizator.
   * **Configurarea surselor de date:** Conectează aplicația la diverse surse de date pentru a obține informații actualizate despre cursurile de schimb.
   * **Personalizarea interfeței:** Modifică aspectul și comportamentul aplicației în funcție de preferințele utilizatorilor.
   * **Monitorizarea performanței:** Urmărește performanța aplicației și identifică potențialele probleme.

**Tehnologii Utilizate**

Pentru a construi această aplicație, am utilizat un set de tehnologii moderne și eficiente, precum:

* **Backend:** Node.js, Express.js, MySQL
* **Frontend:** React.js
* **Securitate:** JWT (JSON Web Tokens)
* **Vizualizare:** Grafice și hărți interactive

**Beneficii**

* **Accesibilitate:** Oferă informații precise și actualizate despre cursurile de schimb într-un format ușor de utilizat.
* **Personalizare:** Permite utilizatorilor să își adapteze experiența în funcție de nevoile lor specifice.
* **Automatizare:** Elimină necesitatea de a căuta manual informații despre cursurile de schimb.
* **Decizie informată:** Ajută utilizatorii să ia decizii financiare mai bune, bazate pe date reale.
* **Scalabilitate:** Poate fi ușor extinsă pentru a servi un număr mai mare de utilizatori.

Aplicația noastră web reprezintă un instrument valoros pentru oricine dorește să urmărească evoluția pieței valutare și să ia decizii financiare în cunoștință de cauză. Prin interfața sa intuitivă și funcționalitățile avansate, această aplicație simplifică procesul de monitorizare a cursurilor de schimb și oferă o experiență de utilizare plăcută.

# Modelul Datelor

Platforma dezvoltată este un sistem distribuit bazat pe o arhitectură client-server care separă logic backend-ul (serverul) de componentele frontend. Această separare permite scalabilitate, securitate și o interfață de utilizare performantă. Fiecare componentă a sistemului utilizează protocoale moderne și algoritmi optimizați pentru a asigura funcționarea corectă, manipularea eficientă a datelor și protejarea acestora împotriva accesului neautorizat.

## Aplicația Server / API

Aplicația server reprezintă nucleul sistemului și este responsabilă de gestionarea logicii de business, a interacțiunilor cu baza de date și a comunicării prin API-uri securizate cu aplicațiile frontend.

### Structura bazei de date

Baza de date utilizată în cadrul platformei este *MySQL* și este proiectată să reflecte relațiile logice dintre parteneri, locații, cursurile de schimb și monede. Principalele tabele și relațiile dintre acestea sunt:

* *Partner*: Stochează informații despre parteneri (instituții sau companii) care operează schimburi valutare. Fiecare partener poate avea mai multe locații.
* *Location*: Reprezintă locațiile fizice asociate partenerilor. Fiecare locație poate oferi mai multe schimburi valutare.
* *Rate*: Înregistrează cursurile de schimb valutar pentru diverse locații, cu detalii precum valoarea ratei, moneda de bază și moneda țintă.
* *Currency*: Stochează informații despre monedele disponibile. Monedele sunt configurabile și pot fi adăugate în funcție de cerințele platformei.

### Relații între tabele:

* Un partener (*Partner*) poate avea mai multe locații (*Location*).
* Fiecare locație (*Location*) poate oferi mai multe rate de schimb (*Rate*).
* Fiecare rată de schimb (*Rate*) implică o locatie (*Location*), și o valoare corespunzatoare acesteia.

Această structură relațională asigură flexibilitate și extensibilitate, facilitând gestionarea eficientă a datelor valutare și asocierea acestora cu locații și parteneri specifici.

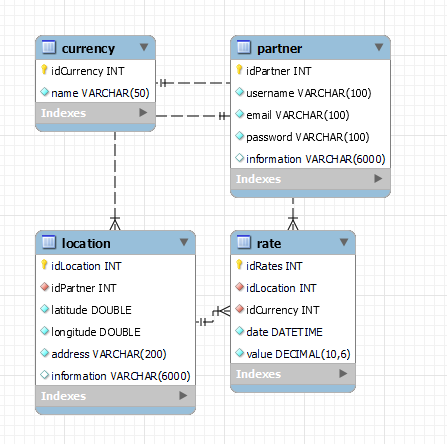


Figura 4.1: Diagrama bazei de date

### Securitatea datelor

Pentru protejarea datelor utilizatorilor de tip administrator și a operațiunilor efectuate în platformă, au fost implementate următoarele măsuri:

* *Criptarea parolelor utilizatorilo*r:
  + Parolele utilizatorilor de tip administrator sunt criptate folosind algoritmul *MD5* înainte de a fi stocate în baza de date.
  + Alegerea *MD5* a fost făcută pentru simplitate și viteza sa de execuție, având în vedere că această platformă are un nivel controlat de utilizatori administratori și nu procesează date critice personale ale clienților finali. Totuși, pentru medii mai stricte, poate fi considerată trecerea la algoritmi mai moderni, cum ar fi bcrypt.
  + Avantajul criptării este că niciun utilizator rău-intenționat nu poate obține direct parolele chiar dacă accesul la baza de date ar fi compromis.
* *Endpoint-uri securizate:*
  + Accesul la API-urile aplicației server este protejat prin utilizarea *JWT*(JSON Web Tokens).
  + *JWT* este configurat cu o cheie secretă unică, care este stocată în mod sigur și utilizată pentru a semna și valida token-urile. Aceasta permite autentificarea și autorizarea utilizatorilor, asigurând că doar cererile validate au acces la resursele API-ului.
  + Toate cererile către API necesită un token valid pentru a accesa endpoint-urile securizate, reducând riscul de acces neautorizat.

# Modelul Arhitectural

Acest capitol prezintă procesul de proiectare și implementare al platformei, detaliind arhitectura modulară și componentele esențiale ale fiecărei aplicații: Server/API, aplicația administrator și aplicația client. Se vor descrie obiectivele, funcționalitățile principale și modul de integrare a acestora, oferind o privire detaliată asupra fiecărui modul în parte, alături de un plan clar pentru implementare.

## Plan de proiectare și implementare

Planul de proiectare urmărește o arhitectură modulară care să permită extinderea și întreținerea facilă a aplicației. Principalele etape ale proiectării sunt:

1. *Definirea cerințelor*: Identificarea funcționalităților esențiale pentru fiecare modul.
2. *Arhitectura aplicației*: Stabilirea unei arhitecturi bazate pe trei niveluri:
   * Backend (server/API) pentru manipularea datelor și securitatea accesului.
   * Frontend-ul aplicației administrator pentru gestionarea resurselor platformei.
   * Frontend-ul aplicației client pentru oferirea unei experiențe interactive utilizatorilor finali.
3. *Implementarea modulară*: Divizarea fiecărei aplicații în componente autonome pentru reducerea complexității.
4. *Testare și integrare*: Asigurarea unei funcționări corecte prin testări unitare și de integrare.

A screenshot of a computer screen

Description automatically generated

Figura 5.1: Diagrama fluxului de lucru

## Aplicația Server / API

* **Descriere generală**

Serverul gestionează accesul la baza de date și implementează logica de business a platformei. Este construit folosind *Node.js* și *Express.js*, cu date stocate într-o bază de date *MySQL*.

* **Funcționalit****ăți cheie**
  + 1. *Managementul partenerilor și locațiilor*

Partenerii pot avea mai multe locații asociate, fiecare locație fiind identificată prin coordonate GPS, adresă și informații suplimentare.

Relația partener-locație este gestionată printr-o legătură între tabelele partner și location.

* + 1. *Configurarea și gestionarea monedelor*

Sistemul permite configurarea unei liste personalizate de monede, fiecare identificată printr-un nume unic.

* + 1. *Adăugarea și administrarea ratelor de schimb valutar*

Ratele de schimb sunt asociate atât unei locații specifice, cât și unei monede. Utilizatorii pot adăuga, actualiza și șterge ratele, respectând constrângerile de integritate.

* + 1. Securitatea datelor

Parolele utilizatorilor sunt criptate folosind algoritmul *MD5*, oferind un nivel de protecție.

Endpoint-urile API sunt securizate cu JSON Web Tokens (*JWT*), utilizând o cheie secretă configurabilă, pentru a garanta accesul exclusiv utilizatorilor autorizați.

* **Structura bazei de date**
  + 1. *Tabela partner*
  1. Coloane:
     + - *idPartner: cheie primară, identifică unic fiecare partener.*
       - *username:* unic, folosit pentru autentificare.
       - *email:* unic, pentru identificare suplimentară.
       - *password:* stochează parola criptată.
       - *information:* detalii adiționale despre partener.
  2. Constrângeri
     + - Toți partenerii au nume de utilizator și email-uri unice.
     1. *Tabela location*
  3. Coloane:
     + - *idLocation: cheie primară, identifică unic fiecare locație.*
       - *idPartner:* cheie străină, face legătura cu un partener.
       - *latitude și longitude:* coordonatele GPS ale locației.
       - *address: adresa* locației.
       - *information:* detalii adiționale despre locație.
  4. Constrângeri
     + - Relația dintre locații și parteneri este definită printr-o cheie străină (*FK\_PartnerLocation*), cu opțiunea de ștergere în cascadă.
     1. *Tabela currency*
  5. Coloane:
     + - *idCurrency: cheie primară, identifică unic fiecare monedă.*
       - *name:* unic, denumirea monedei (ex.: USD, EUR).
  6. Constrângeri
     + - Fiecare monedă are o denumire unică.
     1. *Tabela rate*
  7. Coloane:
     + - *idRates:* cheie primară, identifică unic fiecare curs valutar.
       - *idLocation:* cheie străină, leagă rata de o locație specifică.
       - *idCurrency: cheie străină, leagă cursul de o monedă.*
       - *date: data și ora la care a fost introdus cursul.*
       - *value: valoarea cursului de schimb.*
  8. Constrângeri
     + - Relațiile cu locațiile și monedele sunt definite prin chei străine (*FK\_LocationRate* și *FK\_CurrencyRate*), asigurând consistența datelor.
       - În cazul ștergerii unei locații sau a unei monede, ratele asociate sunt șterse automat.
* **Documentație Swagger**

Aplicația server utilizează *Swagger* (*OpenAPI*) pentru a genera și gestiona documentația API-ului. Swagger oferă o interfață interactivă care permite dezvoltatorilor și utilizatorilor să vizualizeze și să testeze endpoint-urile disponibile direct dintr-un browser web.

Endpoint-urile aplicației server sunt organizate în *module de tip fișier componentă*, fiecare fișier reprezentând o colecție de rute asociate unei funcționalități specifice. Această abordare modulară facilitează organizarea codului, întreținerea și extinderea funcționalităților aplicației.

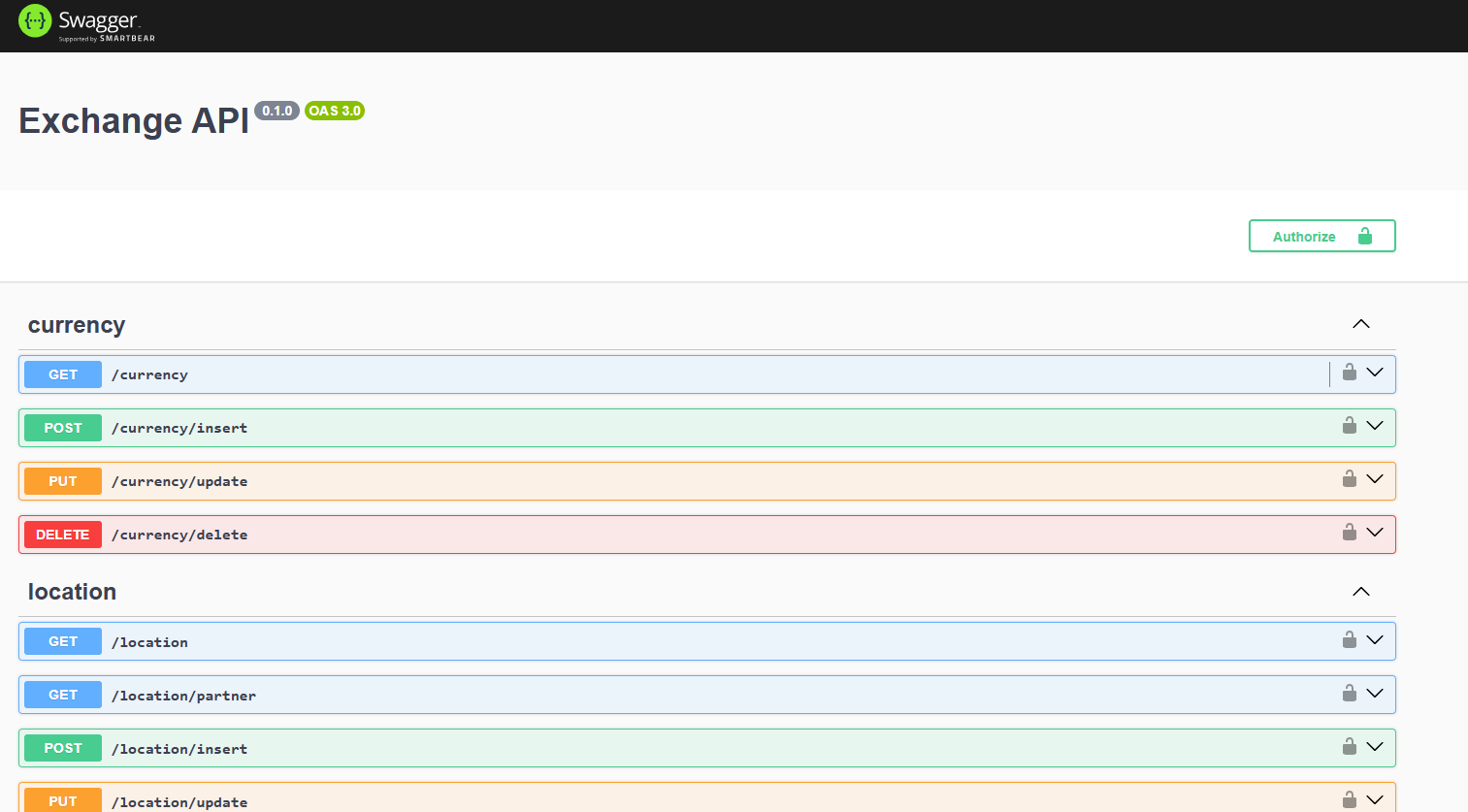


Figura 5.2: Swagger UI pentru API

## Aplicația Administrator

* **Descriere generală**

Aplicația Adminstrator este un *Single Page Application (SPA)* dezvoltată folosind *React.js* și este dedicată gestionării datelor platformei. Aceasta oferă funcționalități precum gestionarea locațiilor, cursurilor valutare și a monedelor însoțite de opțiuni avansate de filtrare pentru o experiență de utilizare optimizată. Aplicația este modulară, organizată pe componente reutilizabile, și integrează notificări pentru operațiunile efectuate, utilizând o navigare bazată pe *React Router.*

* **Structura aplicației**
  + 1. *Pagina de autentificare*
* Oferă acces securizat pentru utilizatorii de tip administrator.
* Parola utilizatorilor este stocată în baza de date folosind algoritmul *MD5*, iar autentificarea este gestionată prin *JWT* pentru securizarea sesiunilor.
  + 1. *Interfața principală*

*După autentificare, aplicația afișează o interfață intuitivă cu un navbar pentru navigare rapidă între module:*

* ***Dashboard***
* ***Locations***
* ***Partners***
* ***Currency***
* ***Rates***
* **Funcționalități cheie**
  1. *Dashboard*

*Prezintă grafice dinamice folosind React Google Charts pentru a vizualiza date statistice despre locațiile și cursurile valutare ale partenerului logat.*

*Datele sunt actualizate în timp real, oferind o privire de ansamblu asupra activităților din platformă.*

* 1. *Locations*
     + Administrarea locațiilor asociate partenerului logat.
     + Adăugare locații noi: Utilizatorul introduce o adresă, iar aplicația generează un preview *Google Maps*.
     + Preluarea automată a coordonatelor GPS: Utilizând API-ul *Google Maps*, se obțin latitudinea și longitudinea corespunzătoare adresei introduse.
     + Gestionarea locațiilor existente: ștergere, actualizare sau modificare informații adiționale.
  2. *Partners*

Afișează o listă cu informațiile publice ale tuturor partenerilor înregistrați în platformă. Informațiile afișate cuprind numele partenerului, adresa de e-mail și detalii suplimentare pentru a uşura comunicarea și colaborarea.

* 1. *Currency*

Gestionarea completă a monedelor:

* *Vizualizare:* Listarea tuturor monedelor existente în platformă.
* *Adăugare:* Introducerea de noi monede.
* *Modificare*: Actualizarea numelor sau altor detalii despre monede.
* *Ștergere:* Eliminarea monedelor care nu mai sunt utilizate.
  1. *Rates*

Gestionarea cursurilor valutare pentru locațiile partenerului logat:

* *Adăugare:* Administratorii pot introduce noi cursuri valutare pentru locațiile proprii, asociate monedelor existente.
* *Vizualizare:* Listarea tuturor cursurilor valutare pentru locațiile configurate.
* *Modificare și ștergere:* Actualizarea sau eliminarea cursurilor existente.
* **Implementare modulară**

Aplicația este organizată în componente React reutilizabile, fiecare componentă fiind responsabilă pentru o funcționalitate specifică. Navigarea între modulele aplicației este realizată prin *React Router*, ceea ce permite afișarea dinamică a conținutului fără a reîncărca pagina.

Exemple de componente:

* + *Dashboard.js:* Responsabilă pentru afișarea graficelor și statisticilor.
  + *Location*.js*:* Forme și tabele pentru gestionarea locațiilor.
  + *Rates.js:* Gestionarea și afișarea cursurilor valutare.

* **Securitate și UX**
  1. *Autentificare și sesiuni securizate*

Utilizarea *JWT* pentru autentificare și autorizare. Token-ul este stocat în *localStorage* și utilizat pentru verificarea accesului la API-uri.

* 1. *Notificări*

Aplicația utilizează un context global pentru gestionarea notificărilor (de tip *toast*) pentru a afișa mesaje de succes sau eroare în urma operațiunilor efectuate.

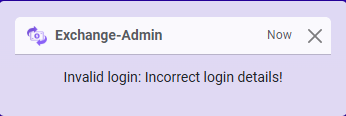


Figura 5.9: Toast notification

* 1. *Design responsive*

Stilizarea aplicației este realizată cu *Bootstrap*, oferind compatibilitate cu diverse dimensiuni de ecran și dispozitive.

## Aplicația Client

* **Descriere generală**

Aplicația Client este o *Single Page Application (SPA)* realizată cu *React.js*, destinată utilizatorilor finali care doresc să vizualizeze cursurile valutare și locațiile disponibile ale partenerilor din platformă. Aplicația oferă o interfață simplă și intuitivă, care permite utilizatorilor să găsească rapid ratele de schimb și să vizualizeze locațiile pe hartă. În plus, aplicația oferă opțiuni de filtrare avansate, bazate pe geolocație sau pe cele mai bune rate de schimb, pentru a facilita experiența utilizatorului.

* **Structura aplicației**

După selecția unui buton, utilizatorii vor fi redirecționați către componentele corespunzătoare care afișează informațiile dorite. *React Router* este utilizat pentru a naviga între componentele aplicației. Modulele aplicatiei:

* + **Nearest Rates**
  + **Best Rates**
  + **All Rates**

Grupare pe locație: Cursurile sunt grupate după locație, astfel încât utilizatorul poate accesa informațiile necesare într-un format ușor de navigat.

* **Funcționalități cheie**
  1. *NearestRates*

A screen shot of a computer program

Description automatically generatedComponenta *NearestRates* este dedicată utilizatorilor care doresc să vadă cursurile valutare ale celor mai apropiate locații. Folosind *geolocația*, aplicația va solicita permisiunea utilizatorului pentru a obține coordonatele GPS și a calcula distanțele față de locațiile de schimb valutar.

Figura 5.11: Raspunsul serverului

* + - *Harta și coordonate*: În această componentă, fiecare locație va fi asociată cu coordonatele GPS (latitudine și longitudine) și va fi vizualizată pe *Google Maps* (acest lucru este recurent și în celelalte componente, deci nu se va relua).

*2.BestRates*

Această componentă afișează cele mai bune rate de schimb pentru fiecare locație disponibilă în platformă. Cursurile valutare sunt ordonate în funcție de valoarea lor, iar utilizatorul poate vizualiza locațiile care oferă cele mai bune rate.

Ratele de schimb sunt filtrate și afișate în funcție de cea mai bună valoare disponibilă, oferind utilizatorului o selecție rapidă și eficientă a celor mai avantajoase opțiuni. Componenta BestRates utilizează sistemul de paginație configurat la nivelul API-ului.

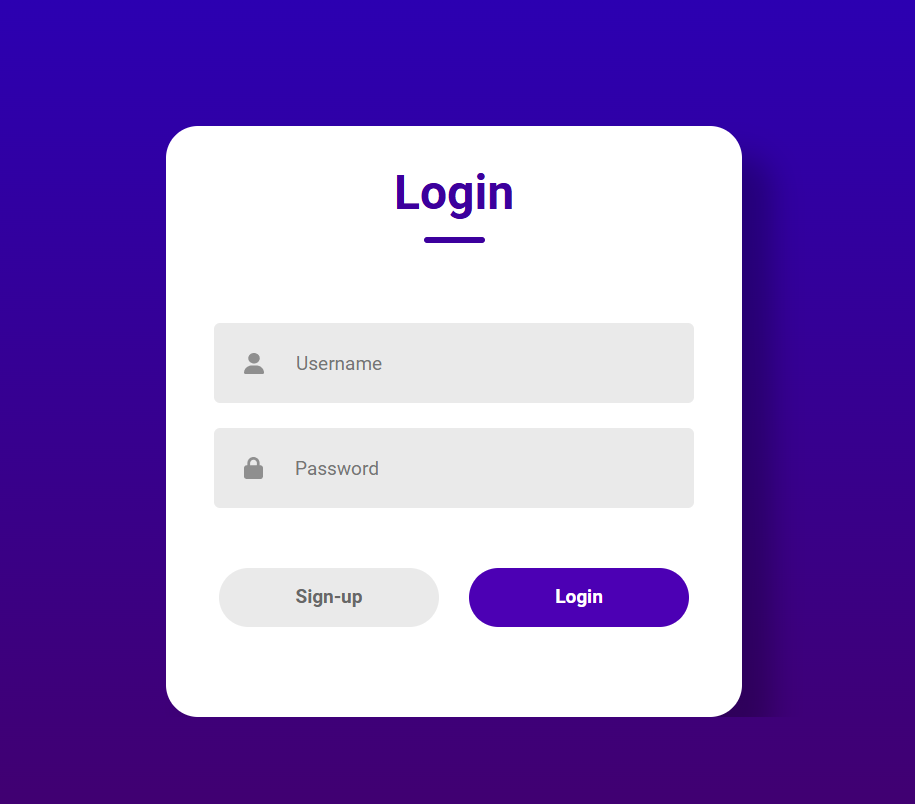
*3.AllRates*

Această componentă listează *toate ratele de schimb disponibile* pentru toate locațiile înregistrate în sistem. Cursurile sunt afișate fără a fi filtrate în funcție de criterii precum distanța sau valoarea, oferind o imagine de ansamblu completă asupra disponibilității și valorii cursurilor valutare în platformă, dar sunt filtrate dupa data inregistrării.

Acest design modular și structura unitară a interfeței garantează o experiență fluidă pentru utilizatori, fie că sunt interesați de cele mai bune rate de schimb, locațiile cele mai apropiate sau o viziune generală asupra tuturor ratelor disponibile.

# Modelul interfetei cu utilizatorul

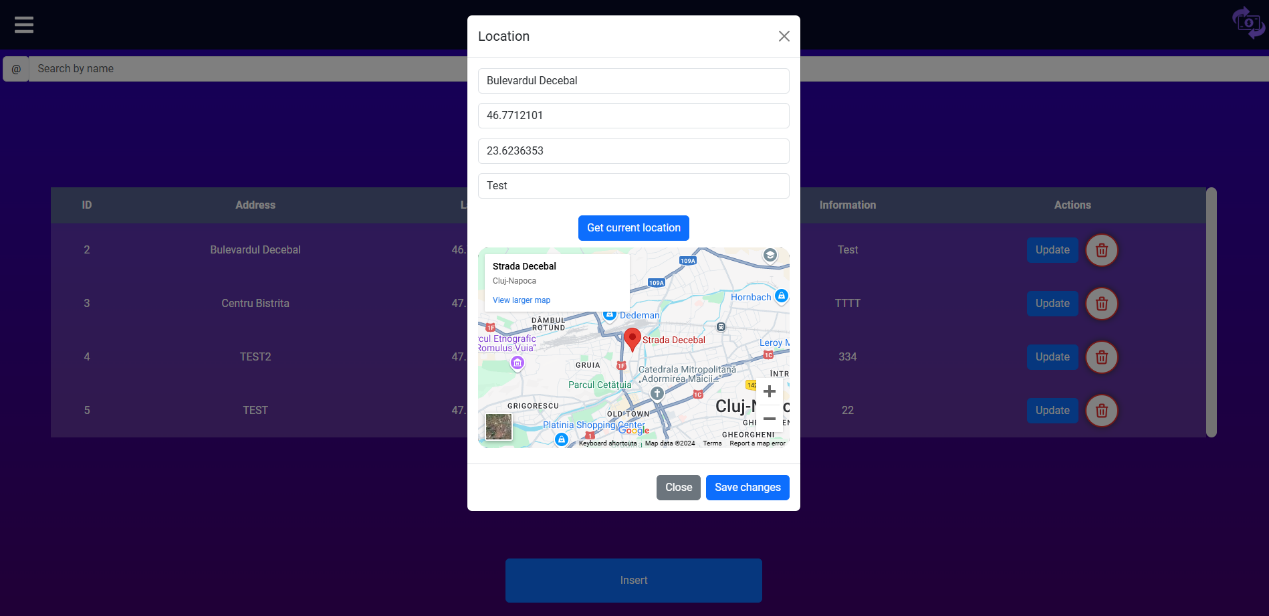
## Aplicația Administrator



UI autentificare



UI dashboard



UI locations

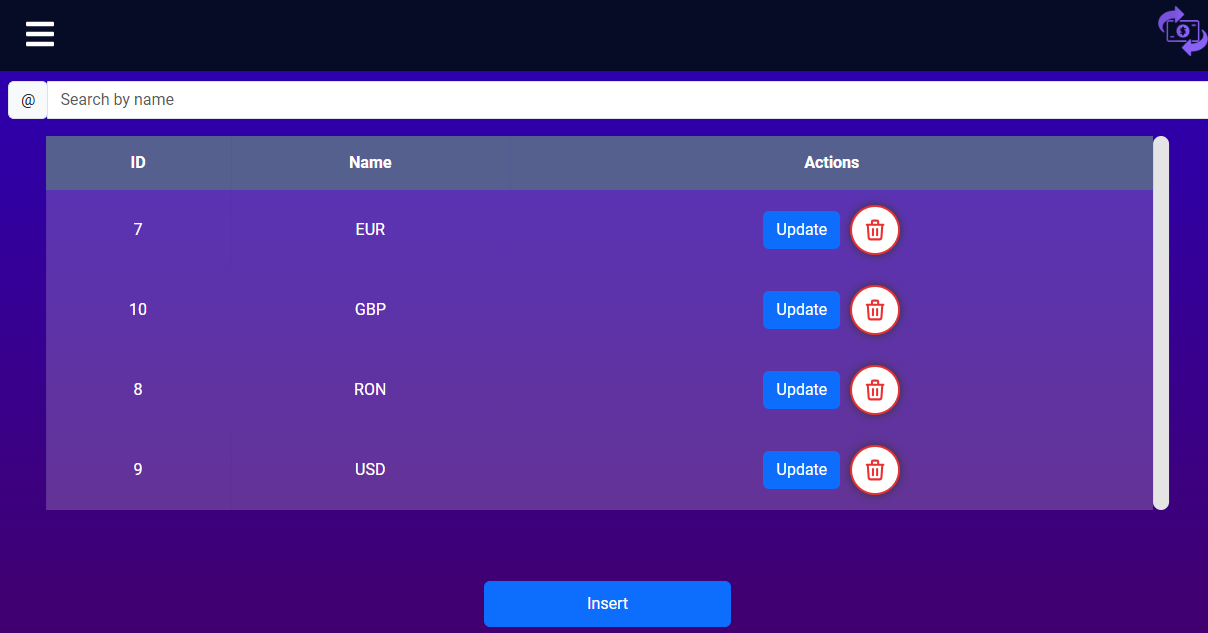
A screenshot of a computer

Description automatically generated

UI partners

A screenshot of a computer

Description automatically generated



UI currency

UI rates

## Aplicatia Client

A screenshot of a screen

Description automatically generated

UI Client

A screenshot of a phone

Description automatically generated

UI Nearest Rates

A screenshot of a phone

Description automatically generated

UI Best Rates

# Capitolul 7. Elementele de testare

Testarea unei aplicații de monitorizare a cursurilor de schimb valutar este esențială pentru a asigura acuratețea datelor, performanța și stabilitatea sistemului. În această secțiune, vom explora câteva dintre cele mai importante probleme de testare pe care le putem întâmpni, cu accent pe scalabilitate, eficiență a datelor și remedierea erorilor tipice.

**Scalabilitate**

* **Creștere a numărului de utilizatori:**
  + Testarea capacității sistemului de a gestiona un număr mare de utilizatori simultani, fără a afecta performanța.
  + Simularea unor vârfuri de trafic pentru a identifica eventuale bottleneck-uri în infrastructura.
* **Creșterea volumului de date:**
  + Evaluarea performanței sistemului pe măsură ce baza de date crește în dimensiune, asigurându-ne că interogările rămân rapide și eficiente.
  + Optimizarea indexării și a structurii tabelare pentru a îmbunătăți performanța căutărilor.

**Eficiență a Datelor**

* **Acuratețea datelor:**
  + Verificarea periodică a datelor din sursele externe pentru a asigura că sunt actualizate și corecte.
  + Implementarea unor mecanisme de verificare a consistenței datelor în baza de date.
* **Viteza de actualizare:**
  + Testarea timpului necesar pentru a actualiza cursurile de schimb în baza de date și pentru a reflecta aceste modificări în interfața utilizatorului.
* **Optimizarea interogărilor:**
  + Analizarea și optimizarea interogărilor SQL pentru a reduce timpul de răspuns al aplicației.
  + Utilizarea de cache pentru a stoca rezultatele interogărilor frecvente.

**Bug-uri Tipice și Scenarii de Testare**

* **Erori de autentificare:**
  + Testarea diferitelor scenarii de autentificare (utilizator valid, utilizator invalid, parola greșită).
  + Verificarea mecanismelor de securitate pentru a preveni atacurile de tip brute force.
* **Erori de afișare:**
  + Verificarea corectitudinii datelor afișate în interfața utilizatorului, inclusiv grafice și tabele.
  + Testarea diferitelor rezoluții ale ecranului și browsere.
* **Erori de conversie valutară:**
  + Verificarea acurateței calculelor de conversie valutară pentru diferite perechi de valute și sume.
* **Erori de notificare:**
  + Testarea funcționalității de notificare pentru alerte și actualizări.
  + Verificarea livrării corecte a notificărilor prin e-mail sau push.
* **Erori de performanță sub sarcină:**
  + Simularea unor scenarii de utilizare intense pentru a identifica eventuale probleme de performanță, cum ar fi întârzieri în încărcarea paginilor sau erori de timeout.
* **Testare de securitate:**
  + Identificarea și remedierea vulnerabilităților comune, cum ar fi injecțiile SQL, XSS și CSRF.
  + Testarea penetrării pentru a evalua nivelul de securitate al aplicației.
* **Testare de compatibilitate:**
  + Verificarea funcționalității aplicației pe diferite browsere și sisteme de operare.
* **Testare de utilizabilitate:**
  + Evaluarea ușurinței în utilizare a interfeței utilizatorului prin teste cu utilizatori reali.

Testarea unei aplicații de monitorizare a cursurilor de schimb valutar este un proces continuu și complex. Prin abordarea sistematică a acestor aspecte, putem asigura calitatea, fiabilitatea și performanța aplicației, oferind utilizatorilor o experiență optimă.

**Recomandări suplimentare:**

* **Automatizarea testelor:** Utilizarea unor instrumente de testare automatizate pentru a accelera procesul de testare și a reduce erorile manuale.
* **Testare continuă:** Integrarea testelor în procesul de dezvoltare pentru a identifica și remedia problemele cât mai devreme posibil.
* **Monitorizarea performanței în producție:** Utilizarea unor instrumente de monitorizare pentru a identifica și rezolva problemele care apar în mediul de producție.

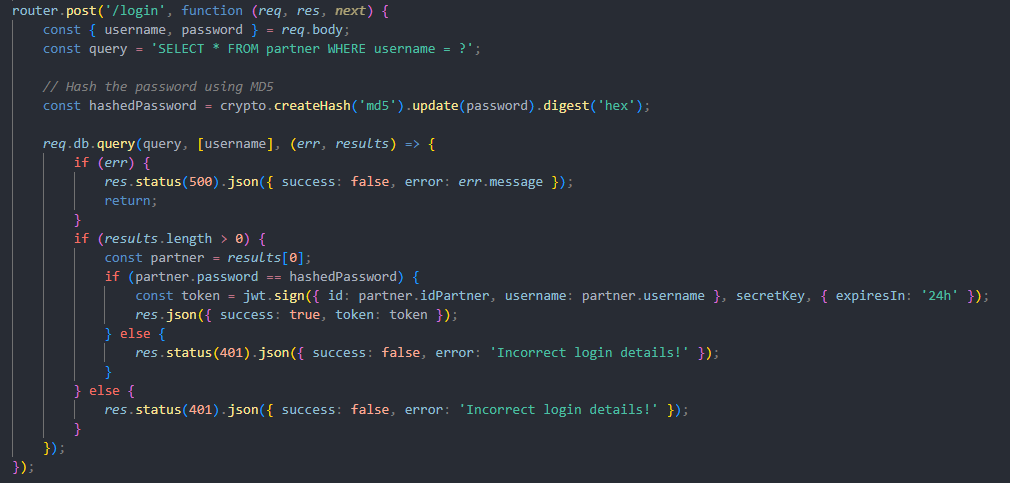
Prin adoptarea unei abordări riguroase de testare, putem construi o aplicație de monitorizare a cursurilor de schimb valutar care să satisfacă cele mai înalte standarde de calitate.

# Anexe

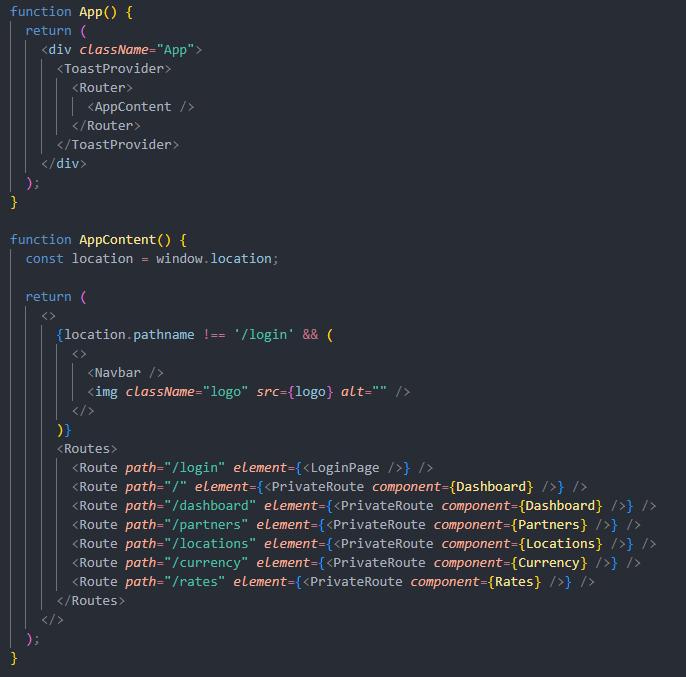
* **Secvențe de cod relevante**:



Anexa 1: Endpoint API pentru client, pentru cursuri în funcție de distanță



Anexa 2: Endpoint securizat din API, pentru logarea administratorilor



Anexa 4: Navigare cu React Router pentru aplicația administrator



Anexa 3: Toast Context

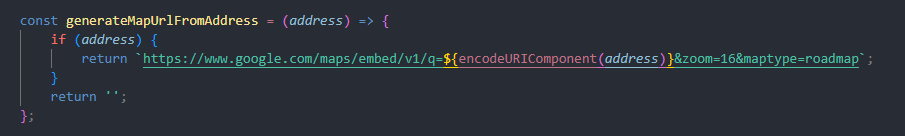


Anexa 5: Implementarea React Table

A screen shot of a computer program

Description automatically generated

Anexa 7: Cod UI Client



Anexa 6: Generare URL pentru Google Maps