UPB Zadanie 5: Dokončenie aplikácie

Autori: Bc. Miloš Ilovský, Bc. Jakub Chrappa, Bc. Marek Mikula, Bc. Boglárka Farkas, Bc. Dávid Kurek

1. **Rozšírenie databázového modelu:**

Prvý krokom pri práci na tomto zadaní bolo rozšírenie našej databázovej štruktúry pre potreby tohto zadania. Ku už existujúcemu používateľovi „user“ sme pridali cesty, ktoré si môže používateľ prezerať a vyberať („route“, táto tabuľka je aj naplnená obrovským množstvom použiteľných dát) a takisto faktúry, ktoré obdrží po kúpe cestovných lístkov na dané cesty („invoice“):

Cesty („route“):

```
CREATE TABLE
  "route" (
    id int PRIMARY KEY NOT NULL,
    type varchar(50) NOT NULL,
    city varchar(50) NOT NULL,
    departure_city varchar(50) NOT NULL,
    arrival_city varchar(50) NOT NULL,
    departure_date date NOT NULL,
    departure_time time NOT NULL,
    arrival_date date NOT NULL,
    arrival_time time NOT NULL,
    price decimal NOT NULL
  );

COPY "route" (
  id,
  type,
  city,
  departure_city,
  arrival_city,
  departure_date,
  departure_time,
  arrival_date,
  arrival_time,
  price
)
FROM
  '/docker-entrypoint-initdb.d/02_routes.csv' DELIMITER ',' CSV HEADER;

```

Naplnené dáta (Ukážka z PgAdmin):

	id [PK] integer	type character varying (50)	city character varying (50)	departure_city character varying (50)	arrival_city character varying (50)	departure_date date	departure_time time without time zone	arrival_date date	arrival_time time without time zone	price numeric
3448	3448	Plane	Brussels	Brussels	Lisbon	2024-05-14	12:00:00	2024-05-14	14:08:26	137.7
3449	3449	Plane	Lisbon	Lisbon	Vienna	2024-05-14	18:00:00	2024-05-14	20:52:23	220.12
3450	3450	Plane	Dublin	Dublin	Brussels	2024-05-15	08:00:00	2024-05-15	08:58:09	56.18
3451	3451	Plane	Rome	Rome	Lisbon	2024-05-15	12:00:00	2024-05-15	14:19:43	53.85
3452	3452	Plane	London	London	Lisbon	2024-05-15	18:00:00	2024-05-15	19:58:52	261.44
3453	3453	Plane	Athens	Athens	Rome	2024-05-16	08:00:00	2024-05-16	09:18:48	234.72
3454	3454	Plane	Lisbon	Lisbon	Rome	2024-05-16	12:00:00	2024-05-16	14:19:43	62.69
3455	3455	Plane	Vienna	Vienna	Athens	2024-05-16	18:00:00	2024-05-16	19:36:12	202.06
3456	3456	Plane	Prague	Prague	Warsaw	2024-05-17	08:00:00	2024-05-17	08:38:46	287.73
3457	3457	Plane	Dublin	Dublin	Rome	2024-05-17	12:00:00	2024-05-17	14:21:25	159.86
3458	3458	Plane	Madrid	Madrid	Budapest	2024-05-17	18:00:00	2024-05-17	20:28:03	104.98
3459	3459	Train	Athens	Athens	Brussels	2023-11-18	05:00:00	2023-11-19	01:53:35	48.44
3460	3460	Train	Dublin	Dublin	Prague	2023-11-18	07:00:00	2023-11-18	21:39:37	28.76

Total rows: 4000 of 267176 | Query complete 00:00:00.699 | Ln 1, Col 1

Faktúry („invoice“):

```sql
CREATE TABLE
  invoice (
    id serial PRIMARY KEY NOT NULL,
    "user" int references "user" (id) NOT NULL,
    invoice bytea NOT NULL,
    paid boolean DEFAULT FALSE NOT NULL,
    created_at timestamp DEFAULT now() NOT NULL
  );
```

V tabuľke (Ukážka z PgAdmin, po už rezervovaní/zakúpení lístkov):

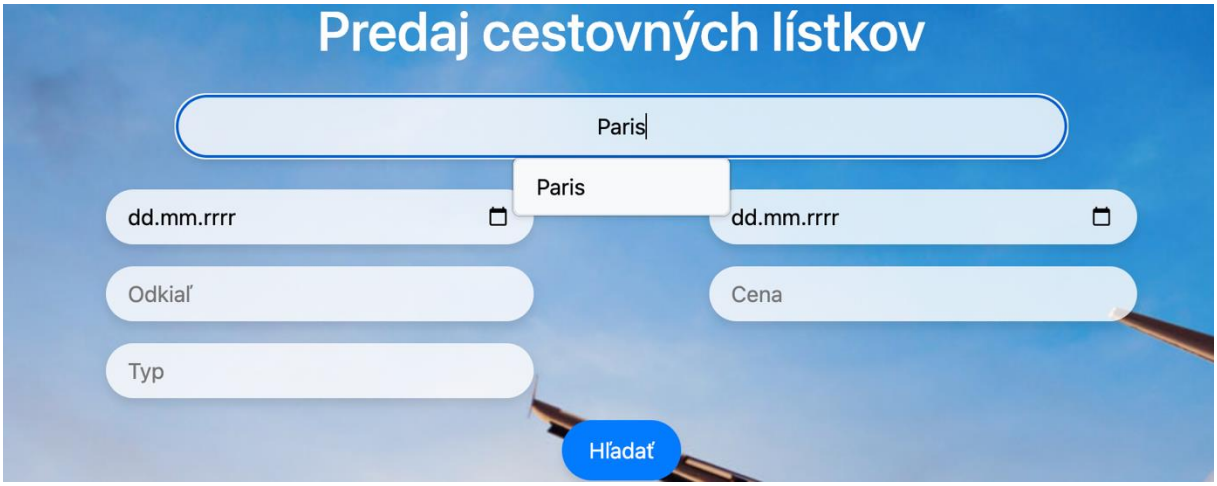
	id [PK] integer	user integer	invoice bytea	paid boolean	created_at timestamp without time zone
1	1	1	[binary dat...	true	2023-11-19 15:07:00.273326
2	2	1	[binary dat...	false	2023-11-19 15:19:10.514411

2. Funkcionality / Vyhľadávacie pole

Pridali sme zvyšné funkcionality. Ku registrácií/prihlasovanie a generovaniu kľúčov sme pridali samotný nákup cestovných lístkov ten je spojený s vyhľadávacím polom, ktoré vyhľadáva nad tabuľkou „route“. Jednoducho povedané prihlásený používateľ napíše kam chce cestovať a zobrazia sa mu všetky cesty do tejto destinácie, následne si používateľ môže danú cestu „kúpiť (zarezervovať)“.

Po tom ako si cestu kúpi (zarezervuje) mu bude odoslaná faktúra, tá sa aj uloží do databázy pod daného používateľa aby k nej mal stále prístup. Faktúru nájde používateľ na svojom profile kde má možnosť faktúru zaplatiť. Taktiež má možnosť aj nezaplatenú aj zaplatenú faktúru stiahnuť (ako PDF).

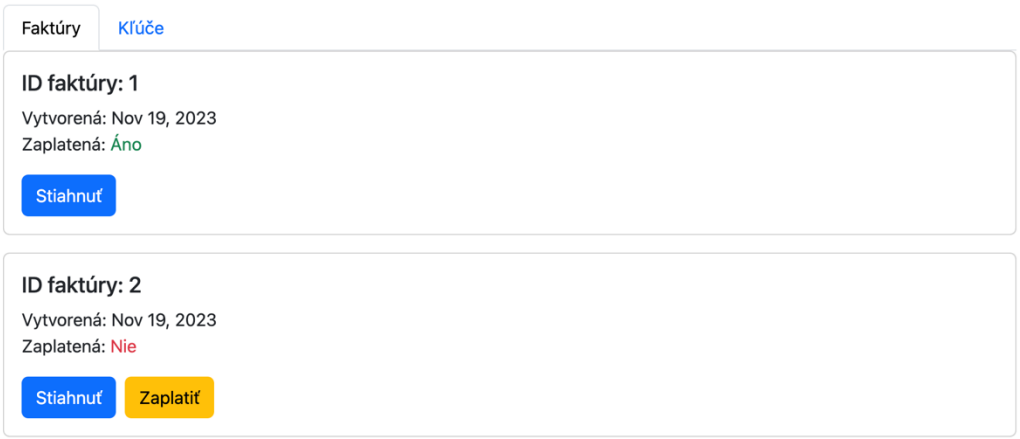
Vyhľadávanie ciest:



List ciest s danou destináciou:

#	Typ	Mesto	Odkiaľ	Kam	Dátum	Čas	Cena (€)	Kúpiť lístok
1	Bus	Budapest	Budapest	Paris	Nov 19, 2023	16:00:00	19.03	Kúpiť
2	Bus	Warsaw	Warsaw	Paris	Nov 21, 2023	14:00:00	49.71	Kúpiť
3	Bus	Oslo	Oslo	Paris	Nov 23, 2023	11:00:00	20.38	Kúpiť
4	Bus	Lisbon	Lisbon	Paris	Nov 23, 2023	12:00:00	15.68	Kúpiť
5	Bus	Dublin	Dublin	Paris	Nov 23, 2023	15:00:00	24.66	Kúpiť
6	Bus	Vienna	Vienna	Paris	Nov 24, 2023	09:00:00	11.29	Kúpiť
7	Bus	Berlin	Berlin	Paris	Nov 24, 2023	14:00:00	40.41	Kúpiť
8	Bus	Warsaw	Warsaw	Paris	Nov 25, 2023	12:00:00	44.78	Kúpiť
9	Bus	Rome	Rome	Paris	Nov 25, 2023	20:00:00	48.17	Kúpiť
10	Bus	Warsaw	Warsaw	Paris	Nov 26, 2023	16:00:00	12.52	Kúpiť
11	Bus	Warsaw	Warsaw	Paris	Nov 27, 2023	06:00:00	42.77	Kúpiť
12	Bus	Berlin	Berlin	Paris	Nov 29, 2023	15:00:00	38.34	Kúpiť
13	Bus	Madrid	Madrid	Paris	Dec 1, 2023	19:00:00	38.72	Kúpiť
14	Bus	Prague	Prague	Paris	Dec 1, 2023	21:00:00	34.27	Kúpiť
15	Bus	London	London	Paris	Dec 2, 2023	13:00:00	15.95	Kúpiť
16	Bus	Lisbon	Lisbon	Paris	Dec 3, 2023	17:00:00	43.14	Kúpiť
17	Bus	Oslo	Oslo	Paris	Dec 4, 2023	17:00:00	13.95	Kúpiť
18	Bus	Oslo	Oslo	Paris	Dec 5, 2023	11:00:00	14.23	Kúpiť
19	Bus	Madrid	Madrid	Paris	Dec 6, 2023	08:00:00	44.24	Kúpiť
20	Bus	Berlin	Berlin	Paris	Dec 6, 2023	20:00:00	32.59	Kúpiť
21	Bus	Warsaw	Warsaw	Paris	Dec 9, 2023	12:00:00	37.28	Kúpiť

Prehľad používateľových faktúr (lístkov):

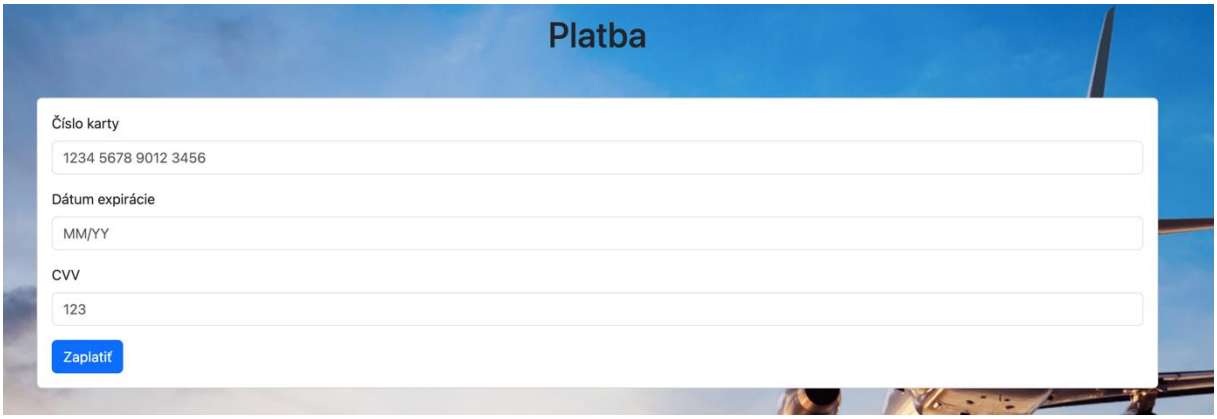


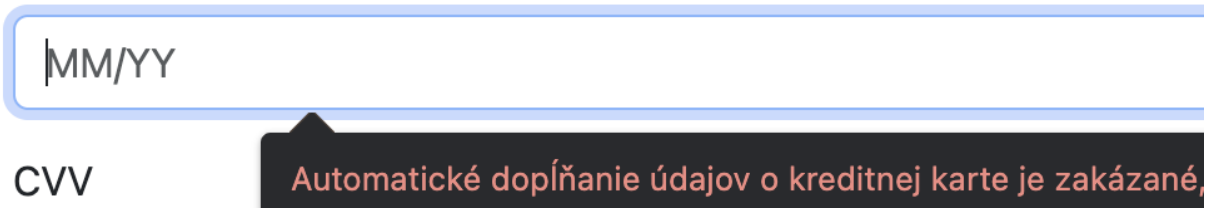
Faktúra PDF:

Name	Jakub Chrappa
E-mail	kubino321@gmail.com
From	Lisbon 2023-11-19 14:00:00
To	Oslo 2023-11-21 11:38:21
Total price	28.11€

Platba faktúr:

Používateľ vyplní informácie podľa šablóny a následne môže zaplatiť faktúru. Faktúra (daná cesta) bude následne evidovaná ako zaplatená a zostane užívateľovi medzi jeho faktúrami (zostane mu možnosť stiahnutia PDF), v databáze bude evidovaná ako zaplatená.





Pri práci na funkcionalitách sme samozrejme dbali na bezpečnosti riešenia a „error handling“ (celé riešenie je súčasťou odovzdaného .zip súboru).

Ukážka error handlingu pri platbe:

```python
invoice = session.exec(select(Invoice).where(Invoice.id == invoice_id)).one()

if invoice is None:
    raise HTTPException(
        status_code=status.HTTP_404_NOT_FOUND,
        detail="Invoice not found",
    )

    # Check if the current user is the owner of the invoice
if invoice.user != current_user.id:
    raise HTTPException(
        status_code=status.HTTP_403_FORBIDDEN,
        detail="You are not authorized to pay this invoice",
    )

    # Check if the invoice has already been paid
if invoice.paid:
    raise HTTPException(
        status_code=status.HTTP_400_BAD_REQUEST,
        detail="Invoice has already been paid",
    )
```

Ukážka bezpečného poslania faktúry:

```
data = io.BytesIO()
PDF.dumps(data, pdf)
data.seek(0)
data = data.read()

invoice = Invoice(
    user=current_user.id,
    invoice=data,
    paid=False,
    created_at=datetime.now(),
)

session.add(invoice)
session.commit()

return Response(
    encrypt(data, current_user.public_key),
    media_type="application/octet-stream",
)
```

Naša encrypt metóda:

```
def encrypt(data, public_key):
    peer_public_key = serialization.load_pem_public_key(public_key)

    key = AESGCM.generate_key(bit_length=128)
    encrypted_key = peer_public_key.encrypt(
        key,
        padding.OAEP(
            mgf=padding.MGF1(algorithm=hashes.SHA256()),
            algorithm=hashes.SHA256(),
            label=None,
        ),
    )

    aesgcm = AESGCM(key)
    nonce = os.urandom(12)
    cipher_text = aesgcm.encrypt(nonce, data, None)

    return encrypted_key + nonce + cipher_text
```

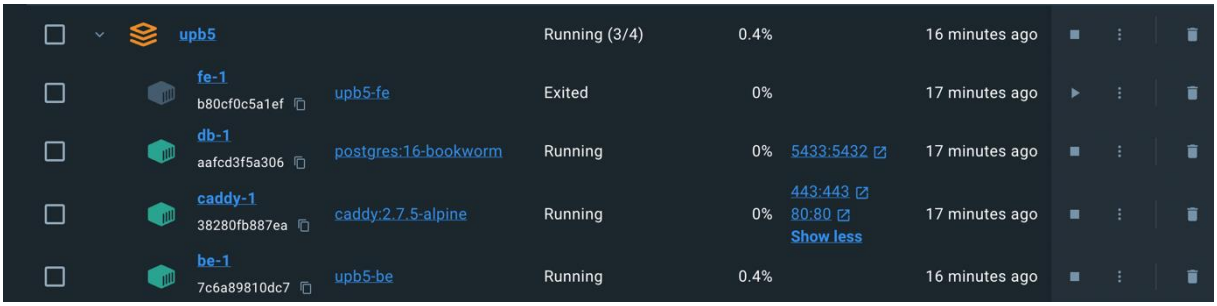
Ku našemu riešeniu sme ako pri každom zadaní vytvorili aj niekoľko testov, ktoré sú súčasťou .zip súboru pod folderom curl_tests v backende:

Data post s implementovaným aj error hadlingom aj encryptovaním dát pomocou public kľúča daného používateľa:

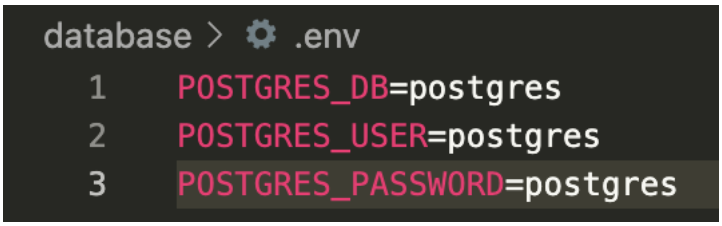
```python
@app.post("/data")
async def get_data(current_user: Annotated[User, Depends(get_current_user)]):
    if current_user.public_key is None:
        raise HTTPException(
            status_code=HTTP_400_BAD_REQUEST,
            detail="User hasn't set up his public key",
        )

    data = json.dumps(
        {
            "first_name": current_user.first_name,
            "last_name": current_user.last_name,
            "mail": current_user.mail,
        },
        sort_keys=True,
        default=str,
    ).encode("utf-8")

    ret = encrypt(data, current_user.public_key)

    return Response(
        ret,
        media_type="application/octet-stream",
    )
```

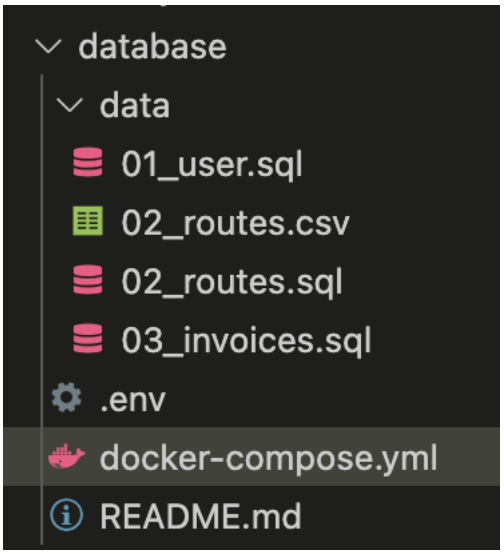
3. Návod na používanie

Aplikácia beží na porte 80:80 cez CaddyServer (stačí docker-compose up --build). Databáza je na porte 5433:5432 :

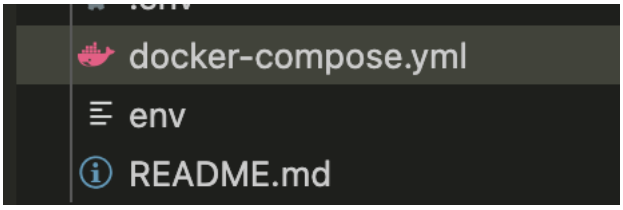


.env súbor potrebný na prácu (basic postgres (súčasť README)):





„.env“ sa musí vytvoriť lokálne je súčasťou zadania ako hidden file. Súčasťou zadanie je aj (pre istotu) „env“ súbor, v prípade potreby ho stačí premenovať na „.env“.

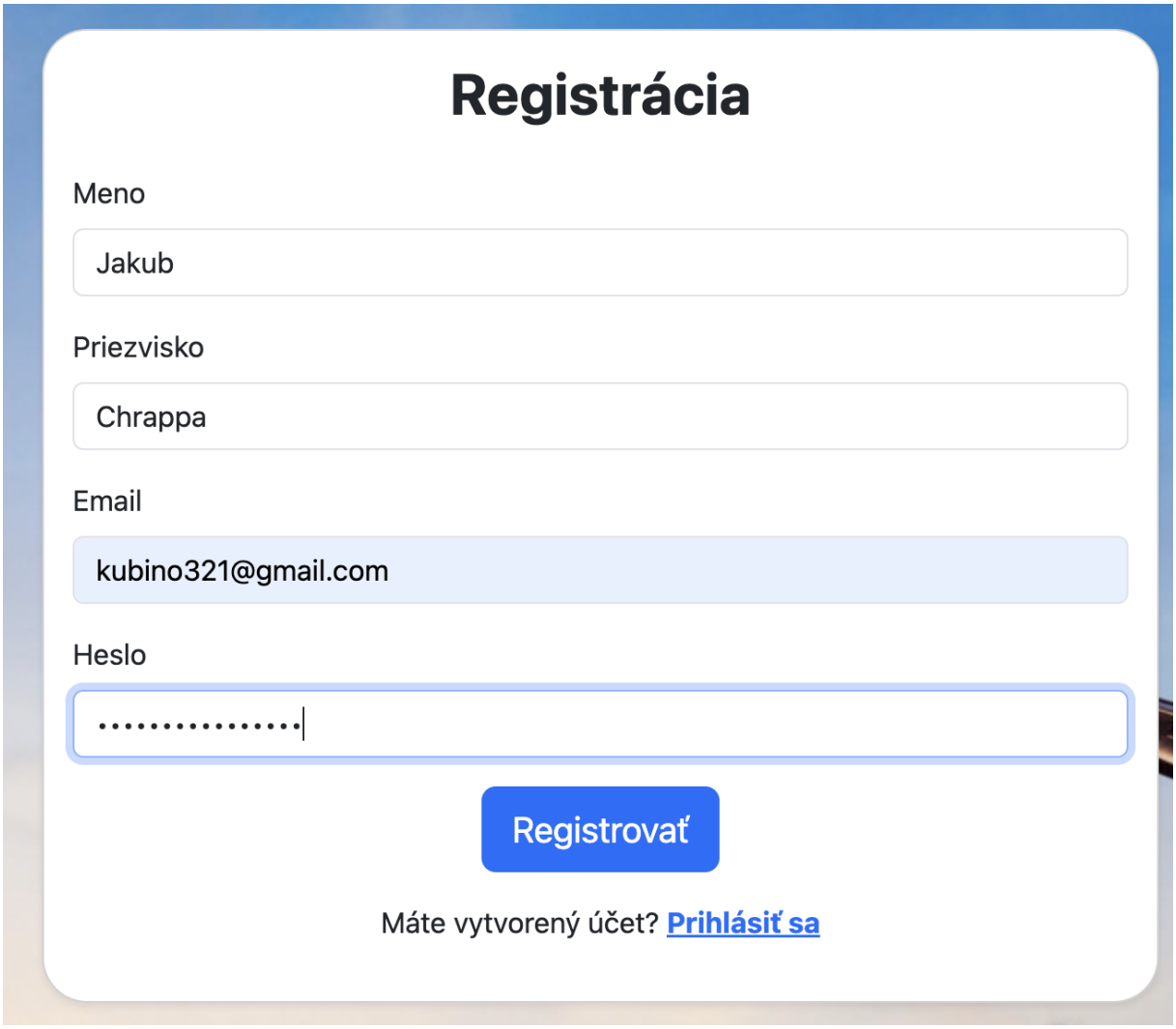


Samotné používanie aplikácie:

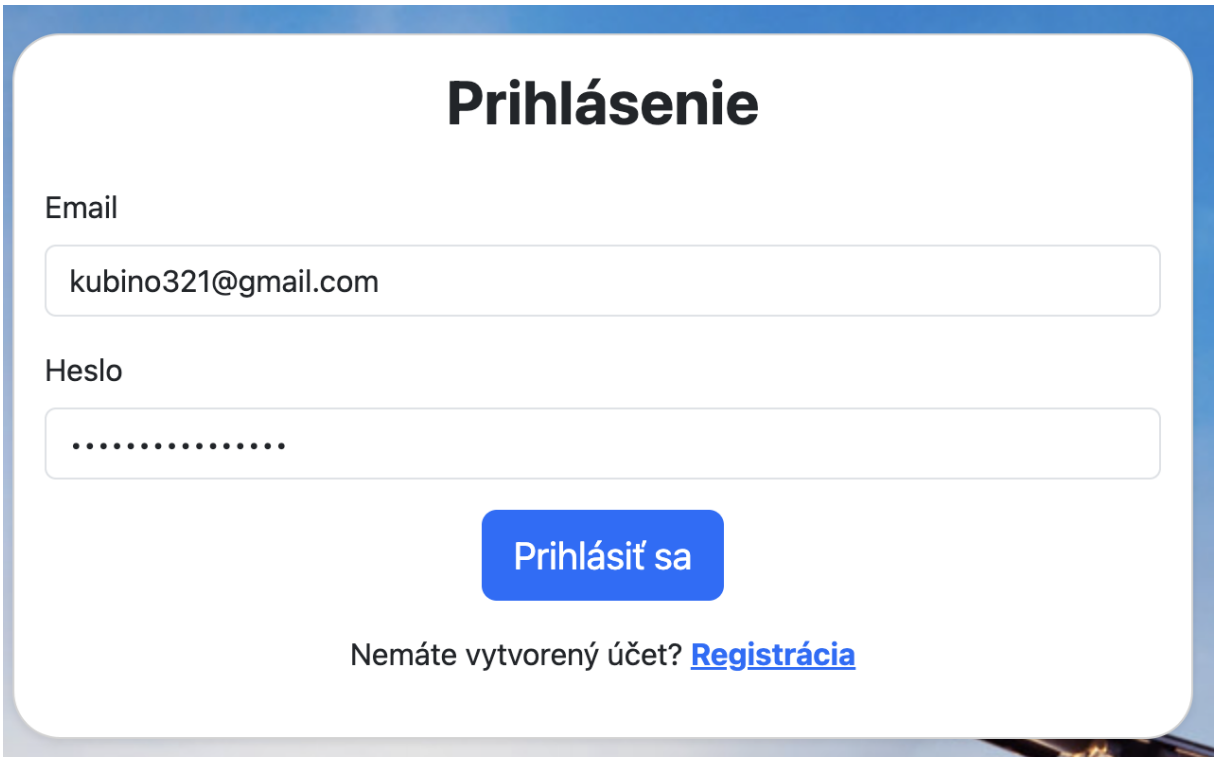
Basic flow: Registrácia, prihlásenie, generovať kľúče, vyhľadať cestu, kúpiť lístok, generovať kľúče (nechceme držať na FE privátny kľúč používateľa pri prechode medzi stránkami z dôvodu bezpečnosti) [odkaz1], prepneme sa na faktúry, dáme stiahnuť, zaplatiť, vložíme údaje a stlačíme zaplatiť. Toto je optimálny flow. Refresh na profile - kľúče si neuchovávame z dôvodu bezpečnosti na FE (public_key má užívateľ v databáze ale nie private) preto treba kľúč vždy vložiť znova.

Basic flow podrobná ukážka:

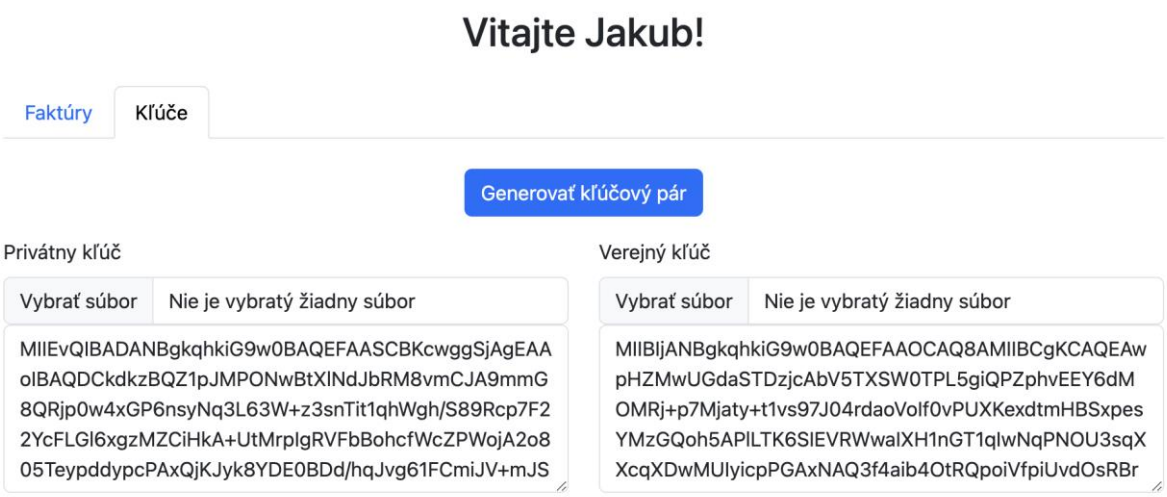
Registrácia:



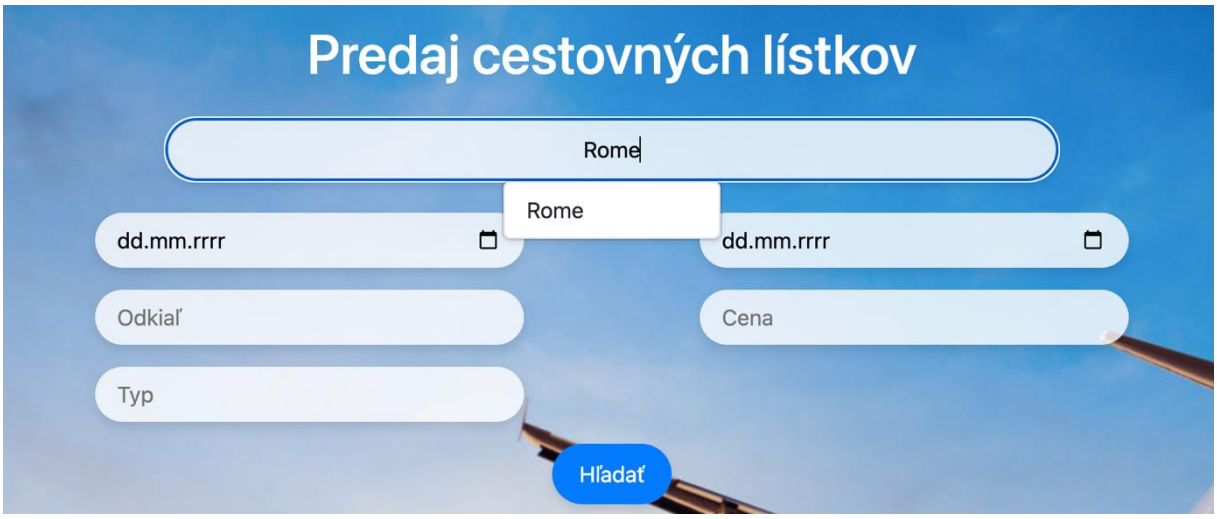
Prihlásenie:



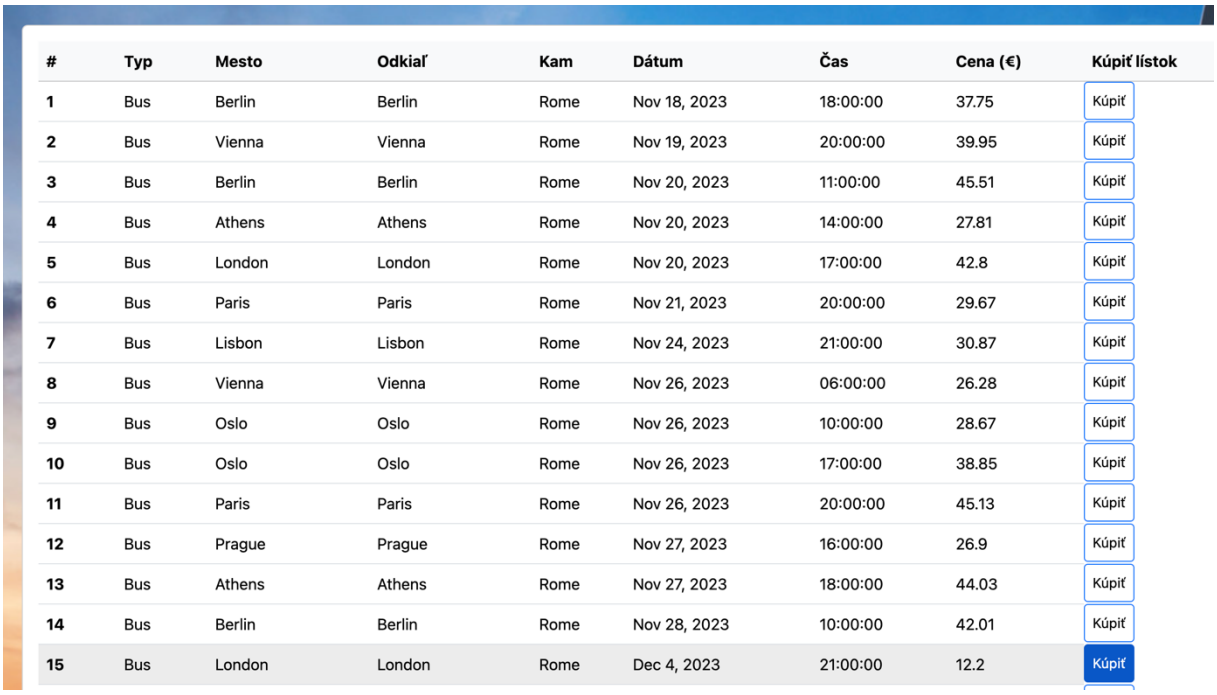
Generovanie kľúčového páru (poprípade nahratie vlastného):



Vyhľadanie ľubovoľnej cesty (záložka UPB)



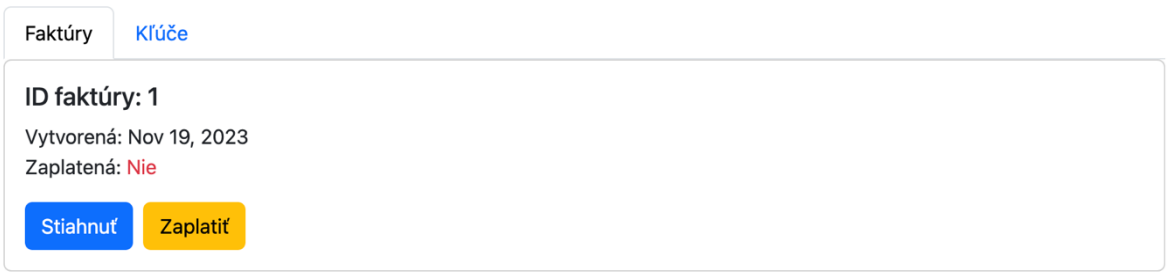
Výber lístka a stlačenie „kúpiť“:



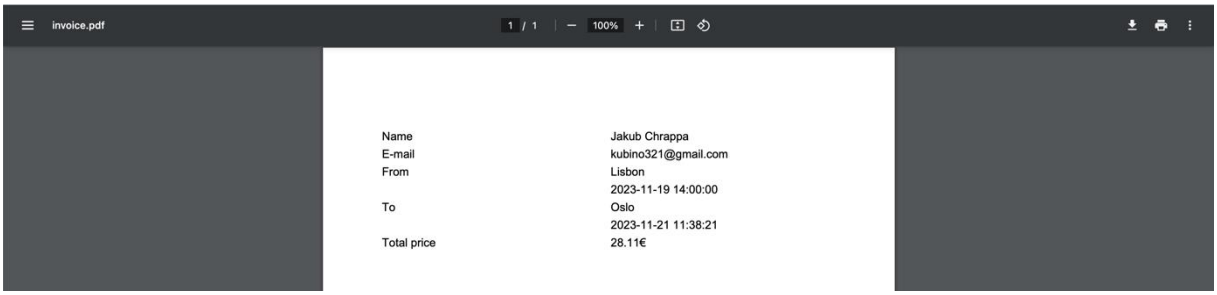
#	Typ	Mesto	Odkiaľ	Kam	Dátum	Čas	Cena (€)	Kúpiť lístok
1	Bus	Berlin	Berlin	Rome	Nov 18, 2023	18:00:00	37.75	Kúpiť
2	Bus	Vienna	Vienna	Rome	Nov 19, 2023	20:00:00	39.95	Kúpiť
3	Bus	Berlin	Berlin	Rome	Nov 20, 2023	11:00:00	45.51	Kúpiť
4	Bus	Athens	Athens	Rome	Nov 20, 2023	14:00:00	27.81	Kúpiť
5	Bus	London	London	Rome	Nov 20, 2023	17:00:00	42.8	Kúpiť
6	Bus	Paris	Paris	Rome	Nov 21, 2023	20:00:00	29.67	Kúpiť
7	Bus	Lisbon	Lisbon	Rome	Nov 24, 2023	21:00:00	30.87	Kúpiť
8	Bus	Vienna	Vienna	Rome	Nov 26, 2023	06:00:00	26.28	Kúpiť
9	Bus	Oslo	Oslo	Rome	Nov 26, 2023	10:00:00	28.67	Kúpiť
10	Bus	Oslo	Oslo	Rome	Nov 26, 2023	17:00:00	38.85	Kúpiť
11	Bus	Paris	Paris	Rome	Nov 26, 2023	20:00:00	45.13	Kúpiť
12	Bus	Prague	Prague	Rome	Nov 27, 2023	16:00:00	26.9	Kúpiť
13	Bus	Athens	Athens	Rome	Nov 27, 2023	18:00:00	44.03	Kúpiť
14	Bus	Berlin	Berlin	Rome	Nov 28, 2023	10:00:00	42.01	Kúpiť
15	Bus	London	London	Rome	Dec 4, 2023	21:00:00	12.2	Kúpiť

Vygenerovať kľúčoví pár / vložiť svoje kľúče znova ako je popísané v basic flow [odkaz1]

Pozrieť faktúru / stiahnuť podľa ľubovôle (Profil -> Faktúry):

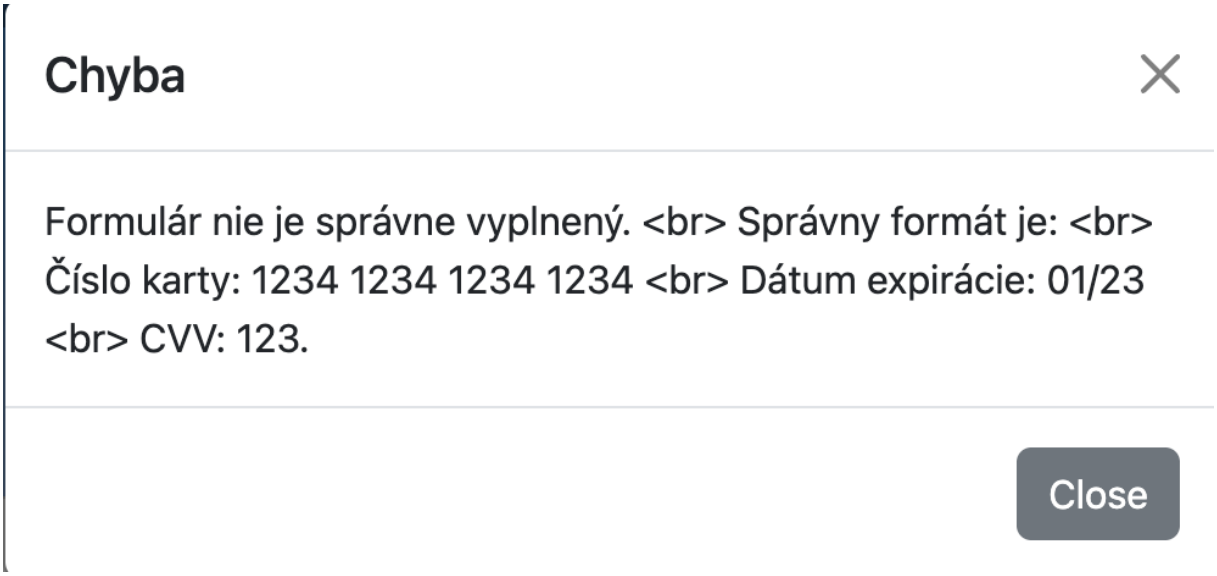


Ukážka stiahnutej faktúry (Ukazuje odkiaľ kam a kedy cestujeme, cenu, meno a mail):

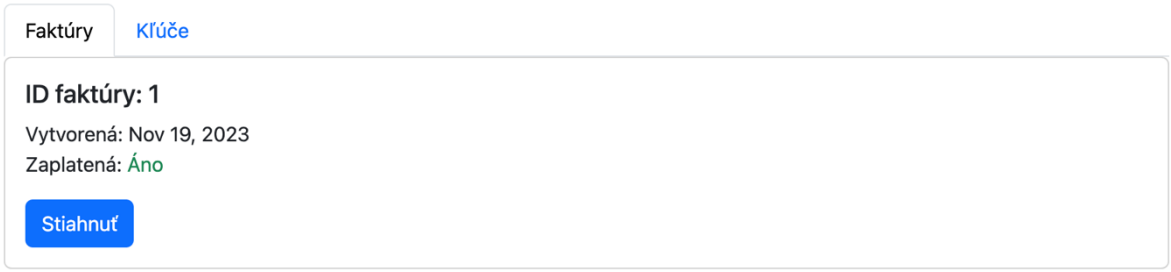


Úhrada faktúry (podľa šablóny):

Dbať na správne vyplnenie údajov aj expirácie (ošetrené v rámci bezpečnosti):



Zaplatiť faktúru („akože“, nebudú strhnuté žiadne peniaze z účtu a žiadne údaje o platobnej karte neuchovávame v databáze):



Toto je podrobný „basic flow“ našej aplikácie, používateľ môže ďalej vyhľadávať / kupovať a prehliadať lístky / faktúry.

Hlasovanie pre Tomáša Vavra ☺: