



Projet JEE

Master Informatique:
Systèmes et Applications Répartis

Résumé du projet:

Prérequis:

- database bankdb
- un serveur mysql
- un navigateur
- Il faut également lancer le MybankAppApplication
- Utilisation de ajax et l'api rest afin de faire communiquer le back-end et le front-end

Fonctionnalités effectuer dans le projet:

- Admin correctement défini (il possède bien tous les droits ,il peut ajouter et supprimer des utilisateurs)
- Les utilisateurs sont bien reliés à des comptes
- Les comptes et utilisateurs sont bien rajouter a la base de donnés
- La connexion par des comptes tiers (google ,github) et créations de comptes sans des comptes tiers
- Déconnexion
- créations de pages html
- créations de projet firebase
- coordination entre le code back-end et front-end

Description:

L'objectif de ce projet est de développer une application simple de services bancaires, en utilisant les technologies JEE et/ou Spring.

Dans un premier temps, afin de pouvoir configurer notre projet, nous avons installé une IDE :Intellij (l'installation de plugin était de pair sachant que maeven était incorporé).

Ensuite afin de pouvoir stocker les clients , nous avons utiliser une base de donnée à l'aide de mysql

Installation :

Commande=>

Mysql Workbench : sudo apt install mysql-workbench

Mysql Community Server

Il faut ensuite Lancer MySQL :

On aura besoin d'une base de données ainsi que qu'un serveur pour pouvoir lancer l'application

service mysql start

mysql -u root -p bankdb

mdp : root

On peut également créer les users et accounts à l'aide de code sql mais en lançant MybankAppApplication (run)

Création de la table users :

```
CREATE TABLE users ( id INT PRIMARY KEY NOT NULL auto_increment=1001, nom VARCHAR(100) not null, prenom VARCHAR(100) not null, email VARCHAR(255) not null, password VARCHAR(255) not null, admin boolean not null default 0);
```

```
alter table users add constraint mail_unique UNIQUE (email);
```

```
CREATE TABLE accounts ( id INT PRIMARY KEY NOT NULL auto_increment, idUser INT foreign key (idUser) references users(id), value double, decouvert int not null default 0, depot int not null default 100, retrait int not null default 100);
```

Ajout d'un utilisateur :

```
INSERT INTO users (nom,prenom,email) VALUES ('ouersighni','haitham','haitham.o@outlook.com');
```

L'id n'est pas à renseigner, il s'auto incrémente.

Dans tous les cas, on lance l'application (soit avec un clic droit run sur MybankAppApplication ou encore ouvrir le fichier et appuyer sur l'icône start)

On lance ensuite sur un navigateur <http://localhost:8080/login> et se connecter pour réussir à se connecter et accéder au site de la banque .

Pour conclure nous avons réussi à créer des utilisateurs ainsi que des comptes reliés à ces derniers à l'aide de class. L'ensemble des utilisateurs et comptes sont défini dans notre projet par les class User ,account et connected qui permettent (en plus des pages html et js) de relier notre back-end et notre front-end via notamment des methodes get et post.

Bonus:

Connexion avec github (authentification):
Il faut avoir npm et firebase pour installer

Dans un premier temps nous avons créé un projet Firebase et récupéré les clés d'authentification afin de les relier avec l'application présente dans le projet Firebase.

Pour votre utilisation:

Ces commandes pourraient vous être utile pour l'utilisation du dossier bonus

```
sudo apt install npm.
```

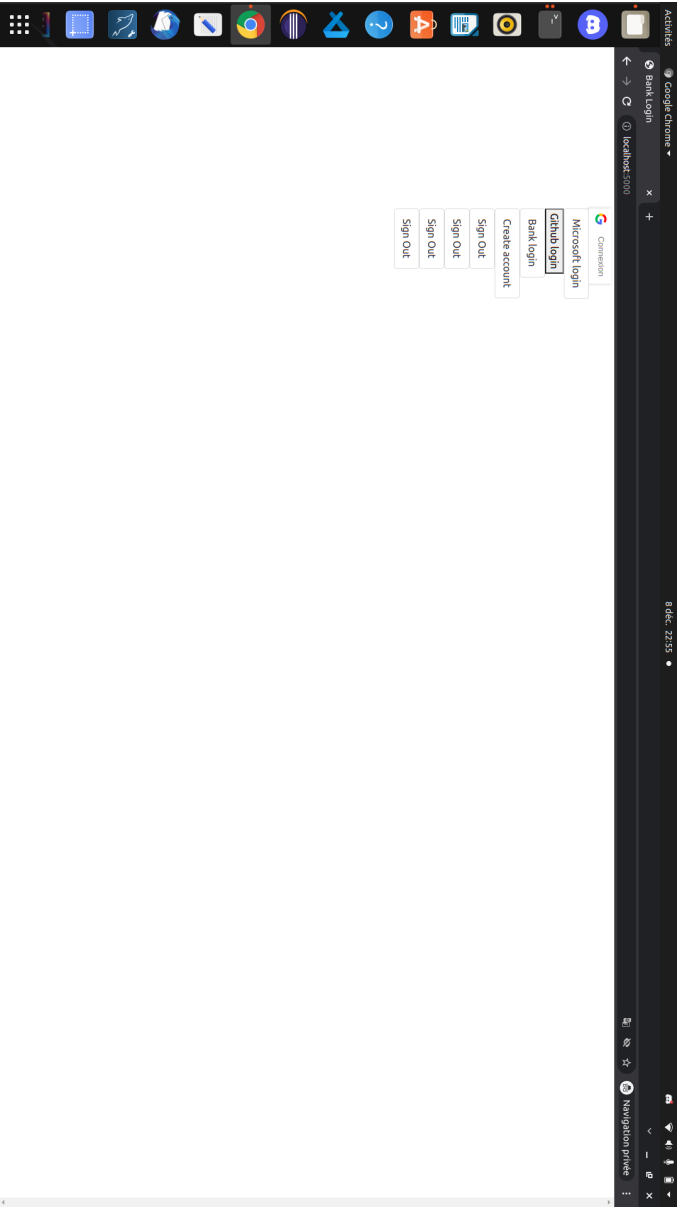
```
npm install -g firebase-tools
```

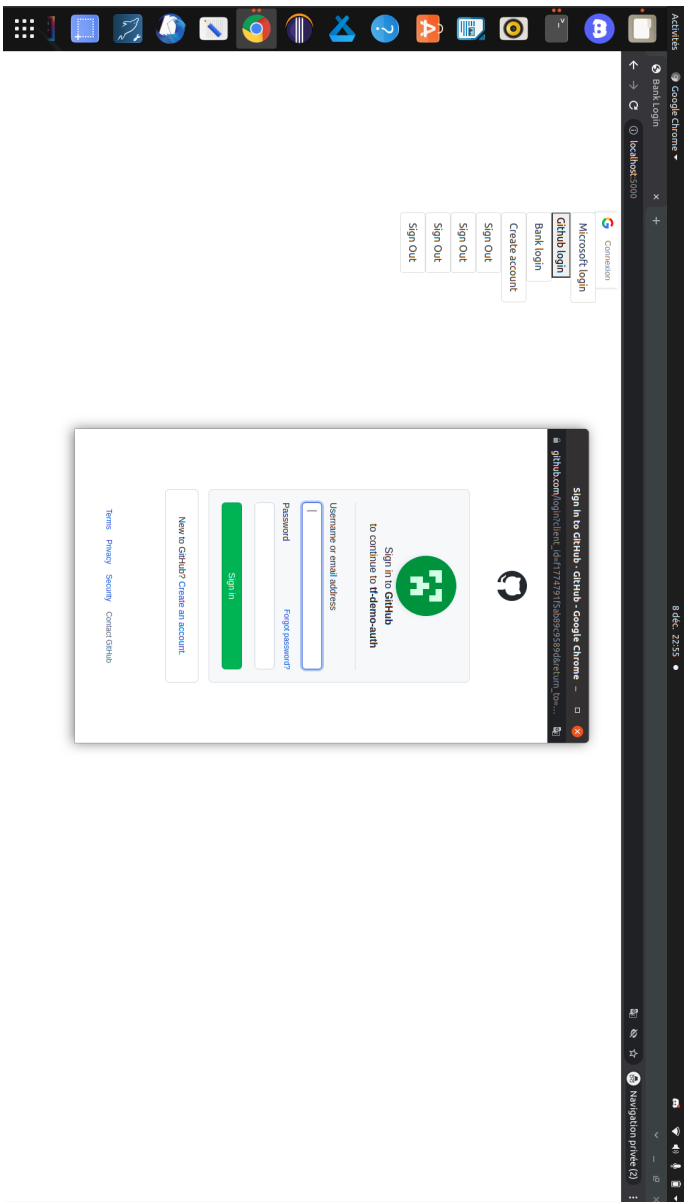
```
sudo apt-get install nodejs
```

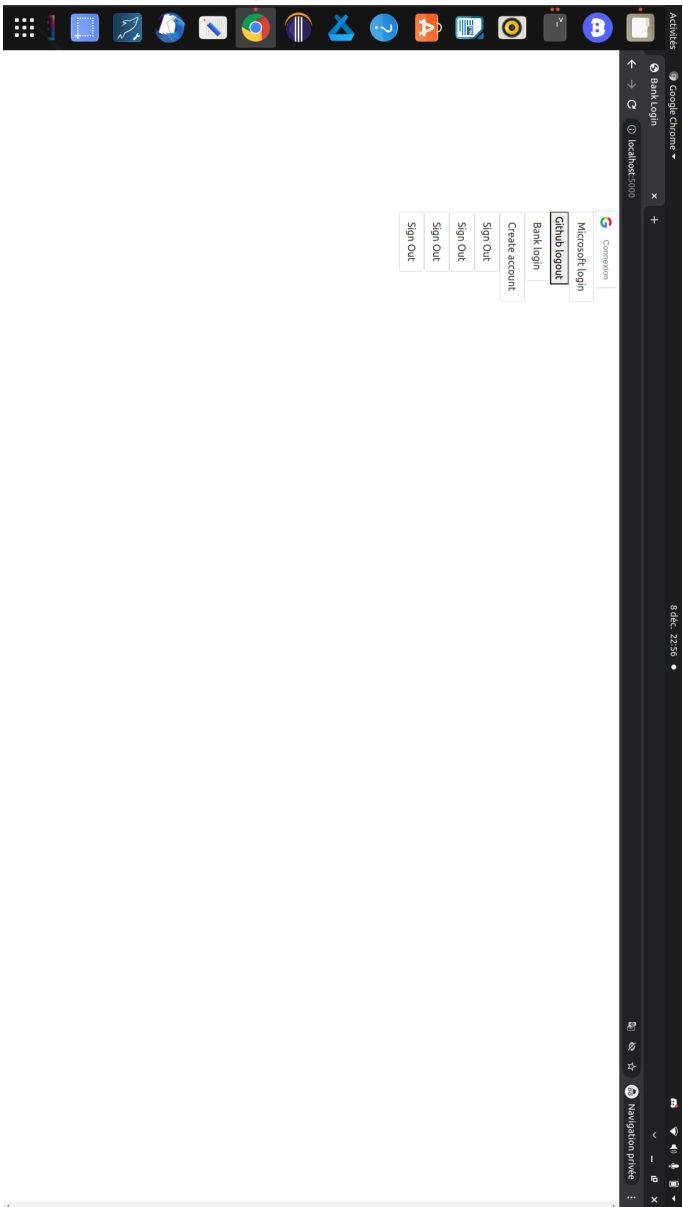
```
npm install -y  
npm init -y  
sudo npm install -g firebase-tools  
firebase init  
firebase login  
firebase init
```

Ensuite il faut run la class MybankAppApplication et lancer sur le <http://localhost:5000>

Voici les résultats attendu:







Fin Bonus