CropCash Documentation

Purpose of this project

Through this project, we want to allow users to rent or buy agricultural machinery with fluid interface.

Team

- BILL AGBOKOU
- YANN-MERIC SOSSOu
- HOUEFA TESSILIMI
- ROY-RONALD HEDIHON-KASSA
- DAVID GBESSEMEHLAN

Project structure

- Frontend interface with react.js + HTML/CSS
- Backend and API management with node.js + express.js
- DataBase with mySQL
- [cropcrash/

```
├── .env
├── package.json
├── server.js
├── public/
│   ├── css/
│   │   └── style.css
│   └── images/        <-- Images des outils et slider
├── views/
│   ├── index.html      <-- Page d'accueil avec slider horizontal
│   ├── register.html    <-- Inscription
│   ├── login.html      <-- Connexion (pop-up en cas d'erreur)
│   ├── profile.html     <-- Profil (affiche les infos de l'utilisateur connecté)
│   ├── tools.html      <-- Liste des outils avec options Buy, Rent, Chat
│   ├── add-tool.html    <-- Pour que les entreprises ajoutent des outils
│   ├── notifications.html  <-- Liste des notifications
```

```
│   ├── chat.html      <-- Page de chat en temps réel avec Socket.IO
│   └── payment.html   <-- Page de redirection après paiement
└── README.md ]
```

Script

npm start and npm run

User Guide

- Customer user choice product
- And process to payment method
- Factory users can propose solutions.

Steps

- git clone https://yourgitrepository
- cd cropcash
- Install dependences with: npm install express dotenv body-parser axios mysql2 multer express-session socket.io
- Create data base and table with following script:
  [CREATE DATABASE IF NOT EXISTS cropcashdb;
  USE cropcash;

  -- Table users
  CREATE TABLE IF NOT EXISTS users (
   id INT AUTO_INCREMENT PRIMARY KEY,
   email VARCHAR(100) UNIQUE NOT NULL,
   password VARCHAR(255) NOT NULL,
   role ENUM('enterprise','farmer') NOT NULL,
   full_name VARCHAR(100),
   phone VARCHAR(50),
   created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
) ENGINE=INNODB;

  -- Table tools

```
CREATE TABLE IF NOT EXISTS tools (
  id INT AUTO_INCREMENT PRIMARY KEY,
  name VARCHAR(100) NOT NULL,
  image_path VARCHAR(255),
  buy_price INT,
  rent_price INT,
  owner_id INT,
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  FOREIGN KEY (owner_id) REFERENCES users(id)
) ENGINE=INNODB;

-- Table payments
CREATE TABLE IF NOT EXISTS payments (
  id INT AUTO_INCREMENT PRIMARY KEY,
  user_id INT NOT NULL,
  tool_id INT NOT NULL,
  amount INT NOT NULL,
  status VARCHAR(50) DEFAULT 'PENDING',
  transaction_ref VARCHAR(100),
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  FOREIGN KEY (user_id) REFERENCES users(id),
  FOREIGN KEY (tool_id) REFERENCES tools(id)
) ENGINE=INNODB;

-- Table notifications
CREATE TABLE IF NOT EXISTS notifications (
  id INT AUTO_INCREMENT PRIMARY KEY,
  user_id INT NOT NULL,
  message VARCHAR(255) NOT NULL,
  is_read TINYINT(1) DEFAULT 0,
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  FOREIGN KEY (user_id) REFERENCES users(id)
) ENGINE=INNODB;

-- Table messages (chat)
CREATE TABLE IF NOT EXISTS messages (
  id INT AUTO_INCREMENT PRIMARY KEY,
  sender_id INT NOT NULL,
```

```
  receiver_id INT NOT NULL,
  message TEXT NOT NULL,
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  FOREIGN KEY (sender_id) REFERENCES users(id),
  FOREIGN KEY (receiver_id) REFERENCES users(id)
) ENGINE=INNODB;
]
```

- Create file in root named ".env"t
- File content [PORT=3000

  DB_HOST=localhost
  DB_USER=root
  DB_PASSWORD=secret
  DB_NAME=agriconnectdb

  INTERSWITCH_BASE_URL=https://qa.interswitchng.com
  INTERSWITCH_CLIENT_ID=YourClientID
  INTERSWITCH_CLIENT_SECRET=YourClientSecret

  APP_REDIRECT_URL=http://localhost:3000/payment-callback]
- API INTEGRATION [{INTERSWITCH_BASE_URL}/passport/oauth/token & {INTERSWITCH_BASE_URL}/paymentgateway/api/v1/paybill]

API REST

- POST /register : Registration (auto-login after registration).
- POST /login : Connection (returns a JSON response for error handling via pop-up).
- GET /logout : Logout.
- POST /profile : Update profile.
- GET /api/profile: Retrieve profile information from logged-in user.
- GET /api/tools: Retrieve tools list (includes owner's email).
- POST /add-tool : Add a tool (accessible to companies only).
- POST /create-bill: Create a bill (Interswitch integration, accessible to farmers only).
- ALL /payment-callback: Payment callback to finalize transaction.
- GET /api/notifications: Retrieve notifications for logged-in user.

- POST /api/notifications/read: Mark a notification as read.
- POST /api/messages: Send a message (chat).
- GET /api/messages: Retrieve the conversation between the logged-in user and another user (via with parameter).

Tests

- Configure your .env file and run the SQL script to create the tables.
- Install dependencies with npm install.
- Launch the server with node server.js.
- Go to http://localhost:3000/ to test the entire workflow (registration, login, adding tools, payment, notifications, chat).
- Open several tabs or browsers to check that the actions (chat, payment, etc.) of one user do not affect those of another.

License