## 🏢 App Description

I want to create a **Windows desktop application it must be packages correctly so i can just import into Github, replit provides workflow and create the exe.** for internal use in an accounting firm.
It should not be a web app — it must run locally and use the internet for syncing data (e.g., via Google Drive). **Electron + SQLite (better-sqlite3)**,

The app will manage **timesheets, clients, and reporting** with **user permissions**.

---

## 👥 USER ROLES

There are two main user types:

1. **Admin User**
    - Can import, edit, and delete timesheets.
    - Can add or edit employee charge-out rates.
    - Can add or edit clients.
    - Can manage permissions for all users.
2. **Normal User**
    - Can only **view timesheets** and **generate reports**.
    - Cannot edit, delete, or create records unless given specific permission.

The admin must be able to select what each user can do: e.g. Edit / Create / Delete / View / Export Reports.

---

## 📄 DATA STRUCTURE

**Clients Table**

- Client ID (Client ID should be displayed in a block on the right top side when pulling PDF reports per client.)
- Client Name (Should be displayed at the top LEFT in a block with pulling PDF reports)
- Notes

**Employees Table**

- Employee ID – This will be used on the timesheets

- Name
- Hourly charge-out rate

**Timesheets Table**

- Date
- Client (linked to Clients table)
- Description of work done
- ID (linked to Employees table or in case of a transfer link the Client ID transferred to )
- Time spent (¼ increments: ¼, ½, ¾, 1, etc.)MUST BE IN FRACTIONS
- Charge-Out amount (auto-calculated)
- Entry Type (Normal / Close-off/Transfer)
- Colour Indicator (normal entries black, close-off entries red, Transfers Green)

Date format dd.mm.yyyy

---

⚙️ **AUTOMATION LOGIC**

 When a timesheet entry is created or imported, the **Charge-Out** field automatically calculates:

Charge-Out = (Employee's hourly rate) × (Time in hours)

 The hourly rate should be fetched dynamically from the **Employees table**.

THE charge out must be rounded off no decimals so if its 375.50 it must be 376.

 If a new employee is added or charge-out rates change, the updated rates are used automatically in new entries, old entries with the old charge out must stay the same.

 The app should allow bulk import of timesheets (from CSV or Excel).

⚙️ **AUTOMATION LOGIC**

**Charge-Out Calculation**

Charge-Out = (Applicable Hourly Rate) × (Time Spent)

- The **Applicable Hourly Rate** is determined based on the **EmployeeRateHistory** table.

- When a new timesheet entry is created, the system looks up the employee's most recent rate **effective on or before the entry date**.
- The **Charge-Out** amount is stored permanently — it never changes retroactively.

**Rate Change Handling (New Rule)**

- When an employee's rate is changed in the **Employees table**, the new rate:
    - **Does not retroactively update** existing timesheet entries.
    - Only applies to **new entries created after the change**.
- The system timestamps each rate change (EffectiveDate).
- The **EmployeeRateHistory** table automatically logs all rate changes.
- During imports:
    - The system checks the timesheet date.
    - If the entry date is **before** the latest rate change, it applies the **older rate**.
    - If after, it applies the **new rate**.

---

💰 **CLOSE-OFF ENTRIES (IMPORTANT)**

- A **close-off** is treated like a **normal timesheet entry**, but:
    - The "Entry Type" = Close-off
    - The **row is displayed in red**.
    - The **Charge-Out** amount is a **negative value** representing what was billed or written off.
    - The close-off entry amount can be **partial** so should be a field where i can add the amount(not necessarily the full total). This will not be a fixed amount and user should be able to enter the amount.
- This allows flexible client reconciliations — you can add multiple close-off entries against one client.

💰 **ENTRY TYPES**

With close off and transfers there isnt a set rate it should be something i can type in as its not charged out rate based

**Normal Entries**

- Entry Type = “Normal”
- Displayed in **Black**
- Standard charge-out calculation.

**Close-Off / Credit Entries**

- Entry Type = “Close-off” – might be other wording must be like a slelection to say its a close off line as the words might differ)
- Displayed in **Red(Words and whole line)**
- Represents billed or written-off work.
- Charge-Out amount stored as a **negative value** (e.g., -R1,500).
- Can be partial or full write-offs.

**Transfer Entries (New)**

- Entry Type = “Transfer”
- Displayed in **Green(Words and whole line)**
- Represents movement of charge-out amounts between clients.
- Requires:
    - Transfer From Client Code
    - Transfer To Client Code
- The amount is deducted from the “From” client and added to the “To” client automatically.
- Reports must show clear audit trails:

e.g. “Transferred R2,000 from Client A001 → Client B004 (Employee: John, Date: 2025-10-16)”

In reports under where employee ID will be the client ID can also be indicated in same column to show which client was transferred to:

Example:

| **CLIENT A** | A1 |
|---|---|

| **DATE** | **DESCRIPTION** | **ID** | **TIME** | **CHARGE OUT** |
|---|---|---|---|---|
| 01.10.2025 | 2/25: AD: FILE WITH CIPC, FILL IN AND SUBMIT FAS | CF | 1/4 | 5313 |

| | | | | |
|---|---|---|---|---|
| 01.10.2025 | GSS CLOSE OFF | | | 3000 |
| 01.10.2025 | TRANSFER TO CLIENT B | B2 | | 2313 |

=0

| CLIENT B | B1 |
|---|---|

| DATE | DESCRIPTION | ID | TIME | CHARGE OUT |
|---|---|---|---|---|
| 01.10.2025 | TRANSFER FROM CLIENT A | A1 | | 2313 |
| 01.10.2025 | 2/25: AD: FILE WITH CIPC, FILL IN AND SUBMIT FAS | CF | 1/4 | 5313 |

=7626

---

## 📊 REPORTS

- Reports by:
  - **Client**
  - **FULL REPORT FOR ALL CLIENTS WHERE EACH CLIENT IS ON THEIR OWN PAGE**
  - **Date Range**
- Reports must show:
  - All timesheet entries (including close-offs)
  - Subtotals for time and charge-out values
  - A separate line or section showing outstanding balances
- Export to Excel and PDF.
- THE COLUMN IN REPORTS AND TIMESHEETS SHOULD JUST BE NAME ID AS WITH NORMAL TIMESHEETS IT WILL SHOW EMPLOYEE ID AND WITH TRANSFERS IT SHOULD SHOW IF ITS A TRANSFER TO IT SHOULD SHOW THE CLIENT ID FOR THE CLIENT IT WAS TRANSFERED TO AND IF ITS TRANSFER FROM IT SHOULD HOW THE CLIENT ID WHERE IT WAS TRANSFERED FROM
- Date format dd.mm.yyyy on exports on pdf and excel
-

---

## 📁 INTERFACE LAYOUT

Tabs or navigation sections:

1. **Clients** – Add/edit clients.
2. **Employees** – Manage employees and charge-out rates.
3. **Timesheets** – View all entries, filter by client, employee, or date, colour-coded rows (red for close-offs).
4. **Reports** – Generate and export reports.
5. **User Management** – Set permissions.

---

## 🔄 SYNCING & STORAGE

- Store all data in a local folder that can sync via **Google Drive for Desktop**.
- Multiple users can access the same synced data file (but editing rights depend on their permissions).
- Allow simple "Backup / Restore" feature (export all data to one file).
- • The system stores its data in a folder that can sync via **Google Drive Desktop Sync**, so multiple users can work on it (with role-based restrictions).
- • If possible, allow local cache so the app can open offline and sync changes later.
- Add a tab or settings screen called "Sync Setup" in the app where users can connect to a shared Google Drive data file so all app installations stay automatically in sync.📁 Step 1: Choose or Create Shared Data File Add a "Select Folder Path" button. When clicked: Let the user choose a folder (typically inside their Google Drive desktop sync folder). If no data file (data.db or data.json) exists in that folder, auto-create one. Save the file path to config.json (or local settings registry). Display a message: "Shared data file created at [path]". 🧍 Step 2: Connect Existing Users Other users open the app → go to Sync Setup → click "Select Existing Data File". They browse to the same file path (Google Drive folder). The path is stored locally and reused on every launch. 🔄 Auto-Sync Mechanism Every few seconds (e.g., 5–10s), or on every database update: Check if the file's last modified timestamp differs from the cached copy. If newer → auto reload new data into the app. If local user changes data → update the file immediately. 🪄 User Interface Example
- Tab Name: Sync Setup
  Components:
- 📂 "Select Folder Path" (creates shared data file)
- 📄 "Select Existing Data File" (connect to existing shared file)
- ✅ Label showing "Current Linked File: C:\Users<Name>\Google Drive\Shared\data.db"
- 🔁 "Sync Now" button

- 🕓 “Last Sync: [timestamp]”
- Toggle for “Auto Sync [On/Off]”

---

## 🗓️ DAILY TIMESHEET IMPORT SECTION

- There must be a dedicated “Import Timesheets” page or button.
- Users can import timesheets for each day from Excel or CSV files.
- Import format (columns):
    1. Date
    2. Client Name
    3. Description
    4. Employee Initials
    5. Time Spent
    6. Entry Type (Normal / Close-off/Transfer)
    7. Charge-Out (optional – if blank, system auto-calculates)
- The system should automatically:
    - Match clients and employees from existing records
    - Create new entries if they don’t exist (optional admin setting)
    - Auto-fill missing charge-outs based on employee rate
    - Mark imported rows clearly (e.g., highlight or note “Imported on 2025-10-16”)
- Import confirmation screen must show summary:
    - Total entries added
    - Errors or missing data
    - Option to cancel or confirm import

## 💾 EXPORTS

- Export any view or report to Excel or PDF.
- Include colour formatting (red close-off rows) in exports.

---

## ✅ ADDITIONAL REQUIREMENTS

- Simple, professional interface (spreadsheet-like view).
- Search and filters for client, date, employee.
- Automatic saving.
- Simple backup function (export all data to one file).
- Support for importing existing Excel timesheets.
- Admin-only "Settings" page for rates, permissions, and Drive folder path.

---

**Final Goal:**
A professional Windows desktop app for accounting-firm timesheet management with full client tracking, charge-out automation, red close-off and green transfer entries, user permissions, reporting, and Google Drive sync support.