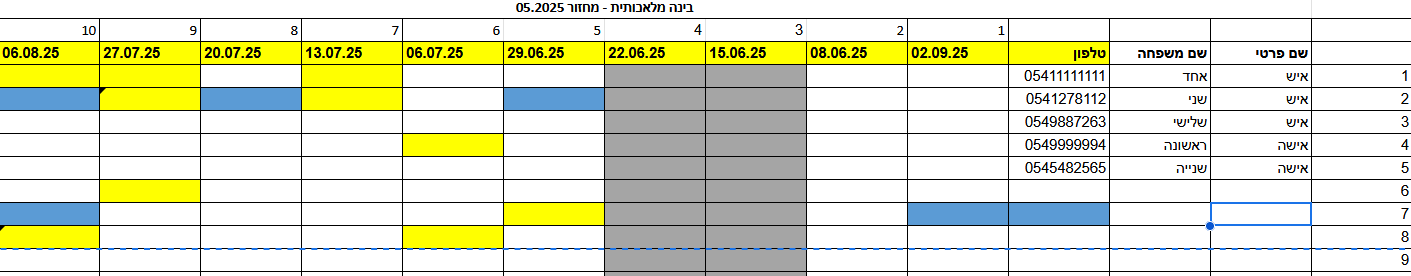
**איך צריך להיראות קובץ האקסל:**

* **קובץ אקסל אחד שמכיל את כל הקורסים לפי טאבים**
* **השם של כל טאב צריך להיות שם התכנית עם מספר מזהה סאפ המתאים**
  + **בעקרון זה לא חייב להיות מזהה סאפ, פשוט צריך מספר אחד עקבי לקורסים. כאשר נוצר קוד QR הוא מקבל את אותו מספר וככה נוצרת ההתאמה**

****

**הטבלה תראה ככה:**

**מה חשוב:**

* **עמודה ראשונה מספרים סידוריים 1,2,3...**
* **עמודה שנייה שם פרטי**
* **עמודה שלישית שם משפחה**
* **עמודה רביעית מספר טלפון** 
  + **מוגדר כטקטס רגיל ולא כמספר (אחרת יעלם ה0)**
* **לוודא שהטלפון בעמודה D, והכותרת עצמה *טלפון* בשורה 3**
  + **כלומר המספר טלפון הראשון יופיע בתא D4**
* **התאריכים צריכים להיות בפורמט כמו בתמונה dd.mm.yy והם נמצאים בשורה 3 בטבלה.**
  + **לדוגמה 01.11.25**

**Full Project Documentation Outline**

**1. Project Overview**

* Project Purpose & Goals
* System Summary & Core Features
* Target Users
* Key Design Decisions
* Tech Stack Overview

**2. System Architecture**

* High-Level Flowchart
* Components Overview
  + QR Generator
  + Web Server (Express)
  + Local Storage ID System
  + PostgreSQL Database
  + Google Sheets API Integration
  + Emergency Backup System
* Data Flow Diagram
* Security Model (Session/Auth)

**3. Installation & Setup**

* Prerequisites
  + Node.js
  + PostgreSQL
  + Python (QR Generator)
  + Google Service Account
* Environment Variables (.env)
* First-Time Setup Instructions
* Folder/File Structure Explanation
* Setting up Google Sheets Access

**4. Frontend (Sign-In Web UI)**

* Sign-In Page Flow
* First-Time User Registration
* UUID Storage (LocalStorage Logic)
* Error Handling & User Messaging
* Token Validation Workflow

**5. Validation Logic**

* Course Validation per Day
* User Matching (Phone → Email → Name)
* Handling Course Overlaps
* Handling Repeated Registrations
* Invalid or Missing Data Paths

**6. Course Scheduling & QR Generation**

* Daily QR Creation Logic
* Token Generation & Expiry
* QR Email Dispatch
* Token-to-UUID Binding
* Fallback if QR Creation Fails
* Example QR Link Structure

**7. Database Design**

* PostgreSQL Tables & Schemas
* How UUID Links to User Info
* Storing Attendance Records
* Archive Logic & Structure
* Migration Notes / Versioning

**8. Google Sheets Integration**

* Format Requirements (Google Sheets vs. Excel)
* Course Sheet Naming Convention
* Participant Sheet Requirements
* Sheet Access Logic
* Error Handling (Not Found, Corrupt, No API Access)

**9. Edge Cases & Recovery Procedures**

* User Has Multiple UUIDs
* User Lost LocalStorage ID
* Switched Devices/Browsers
* Wrong Phone Number in Sheet
* Sheets API Timeout or Failure
* Duplicate Registrations
* Malicious Sign-ins (Token Abuse, Shared Links)

**10. Security Measures**

* Express-Session Overview
* UUID & Token Binding
* Cookie Flags (HttpOnly, SameSite, Secure)
* Token Lifetime & Rotation
* Risks: Manual UUID Deletion, Shared QR Links
* Considerations for Email Authentication

**11. ⚙ Admin & Maintenance Tools**

* Local Admin CLI Tool (Commands List)
* Token Use for API Access
* Generating New Tokens
* Interacting with Server Endpoints
* Logging In & Out (If/when UI added)
* Planned GUI Migration Path

**12. Archiving & Automation**

* Daily QR Automation
* Archiving Past Courses
* Backup Sheet Export Logic
* Handling Archive Failures
* Manual Override Instructions

**13. Notifications & Email System**

* Email Templates
* Email Setup with Nodemailer
* Notification Triggers
  + QR Creation
  + Successful Sign-In
  + Error Alerts
* Emergency Contact Flow (Admin Fallback)

**14. Admin Roles & Permissions**

* Who Has Token Access
* Who Can Generate/Send QR
* Who Can Edit Google Sheets
* Expected Admin Duties
* Knowledge Required for Maintenance

**15. Manuals & How-Tos**

* First-Time Admin Setup
* Daily Workflow
* Manual Sign-In Override
* Updating a Participant’s Info
* Replacing a Corrupted Sheet
* Handling Token Expiry

**16. Testing & QA**

* Test Environment Setup
* Manual Test Cases
* Simulating Sheet Errors
* Token Expiry Testing
* Emergency Scenario Simulations

**17. Disaster Recovery Plan**

* What Happens When Sheets Fail
* What Happens When DB Fails
* How to Restore from Backup
* Emergency Offline Sign-In Flow
* Support Fallback Steps

**18. Future Roadmap**

* Planned Features
* Potential Refactors
* Upgrade Paths (to Firebase, proper DB UI, etc.)
* Performance Improvements
* Security Hardening Plans