**E-Dossier Web Application - Technical Documentation**

**1. Overview**

E-Dossier is a PHP/MySQL web application for managing soldier dossiers, covering personal, family, education, military qualification, physical standards, bank, and punishment details. It provides role-based access for Clerk and CO, data-entry forms, dashboards with charts, tabular views with filters, and printable/exportable reports.

**2. Tech Stack**

- PHP (procedural, mysqli)

- MySQL/MariaDB

- HTML5/CSS3 (Bootstrap 5)

- JavaScript (jQuery, DataTables)

- Chart.js (visualizations)

- jsPDF + autotable (PDF export)

- SheetJS/xlsx (Excel export)

**3. Project Structure (key files)**

- index.php: Public landing dashboard with search and charts

- clerk\_login.php, co\_login.php: Role login pages

- clerk‑auth.php, co‑auth.php: Authentication handlers

- cleark‑dashboard.php, dashboard.php: Role dashboards (Clerk/CO)

- form.html: Data entry form (multi-section)

- insert.php: Persists submitted form to normalized tables

- search‑soldier.php: Renders editable record by Army No

- update‑soldier.php: Updates soldier data across tables

- download‑report.php: Report views + export (PPT/BPET/Firing/Overweight)

- connect.php: Database connection (mysqli)

- logout.php: Session termination

- css: Styles (Bootstrap, custom)

- jschart.js: Chart.js (local for index)

**4. Environments & Setup**

**4.1 Prerequisites**

- PHP  8.x, MySQL/MariaDB, a web server (Apache via XAMPP/WAMP recommended).

- Enable mysqli extension.

**4.2 Database**

- Create database: CREATE DATABASE dossier‑db;

- Configure credentials in connect.php:

$servername = "localhost"; $username = "root"; $password = ""; $dbname = "dossier\_db";

- Ensure required tables exist:

- personal‑details(army‑no PK, rank, trade, name, home‑address, doe, dos, appt, icard‑no, doi, issuing‑auth, medical‑cat, medical‑specific, height, weight, overweight‑percentage)

- family‑details(army‑no FK, father‑name, father‑pt2‑no, father‑pt2‑date, mother‑name, mother‑pt2‑no, mother‑pt2‑date, marital‑status, nok‑details, spouse‑name, dom, pt2‑no, pt2‑date, num‑children, children‑details JSON)

- civil‑qualification(army‑no FK, education‑level, passing‑year, result, university‑name, college‑name)

- military‑qualification(army‑no FK, jn‑cadre, n‑cadre, mr, upgrading‑class, driving‑license, hill‑driving, additional‑course, language‑qualification)

- physical‑standards(army‑no FK, ppt‑date, ppt‑result, bpet‑date, bpet‑result, firing‑date, firing‑result, sports)

- bank‑details(army‑no FK, dsp‑account, joint‑account, bank‑name, loan‑type, loan‑amount, loan‑emi, loan‑duration, loan‑percentage)

- punishment(army‑no FK, red‑ink, black‑ink, punishment)

- users(id PK, username, password MD5, role ENUM('Clerk','CO'))

**4.3 Running Locally**

- Place project under web root (e.g., htdocsdossier).

- Import schema/data into dossier‑db.

- Navigate to http:localhostdossierindex.php.

**5. Authentication & Authorization**

- Roles: **Clerk**and **CO**.

- Login pages: clerk‑login.php, co‑login.php.

- Handlers: clerk‑auth.phpand co‑auth.phpverify userswith MD5-hashed password and set session variables (user‑id, username, role).

- Redirects: Clerk ­> cleark‑dashboard.php, CO ­> dashboard.php.

- Password change: clerk‑change‑password.php, co‑change‑password.php(verifies old MD5, updates to new MD5).

**6. Core User Flows**

**6.1 Public Dashboard (index)**

- Aggregates physical‑standardsfor PPT/BPET/Firing charts.

- Search by Army No forwards to role dashboards or detail views.

**6.2 Clerk Data Entry**

- Navigate to form.htmlto capture soldier data across sections.

- On submit, insert.phpvalidates/normalizes and inserts into all tables within a transaction.

**6.3 Search & Edit**

- search‑soldier.phprenders an editable composite form when provided army‑no.

- update‑soldier.phpupdates the corresponding records across tables.

**6.4 Dashboards**

- cleark‑dashboard.php: CO-like summary with charts and a comprehensive table of soldiers.

- dashboard.php: CO Dashboard with charts and DataTables view.

**6.5 Reports & Exports**

- download‑report.phpoffers dynamic reports: **ppt**, **bpet**, **firing**, **overweight**.

- Client-side exports: Excel via SheetJS and PDF via jsPDF autotable.

**7. Pages and Responsibilities**

- index.php: Landing charts (Chart.js), links to role logins.

- clerk‑login.php, co‑login.php: Styled login forms.

- clerk‑auth.php: Validates Clerk, starts session, redirects to cleark‑dashboard.php.

- co‑auth.php: Validates CO, redirects to dashboard.php.

- cleark‑dashboard.php: Charts + wide DataTable (joined soldier data).

- dashboard.php: Charts + DataTable for CO.

- form.html: Multi-section data-entry with dynamic children fields and conditional spouse fields.

- insert.php: Inserts normalized records in a transaction.

- search‑soldier.php: Loads soldier by Army No and renders input fields for editing across categories.

- update‑soldier.php: Executes UPDATEs across all related tables.

- download‑report.php: Report table with search, date filtering (where applicable), Excel/PDF export.

- logout.php: Destroys session and redirects to login.

**8. Data Model Notes**

- One-to-one logical partitioning by category using army‑noas the join key.

- family‑details.children‑detailsstores JSON array of child objects name, dob, pt2no, pt2date.

- physical‑standards.sportsstores comma-separated values from checkbox list.

**9. Security Considerations**

- Password hashing uses MD5; migrate to password‑hash()/password‑verify() for stronger security.

- Use prepared statements everywhere; some pages build dynamic SQL strings for UPDATEs. Prefer parameterized updates to prevent injection.

- Validate and escape all user inputs both server-side and client-side.

- Consider CSRF tokens for POST forms (insert/update/password change).

- Restrict access to role dashboards and update endpoints via session checks.

**10. Frontend Behavior**

- Styling: Bootstrap 5 + custom CSS, camo background, Poppins/Luckiest Guy fonts.

- Tables: DataTables with paging, search, horizontal scroll where needed.

- Charts: Chart.js doughnut/bar/pie for PPT/BPET/Firing distributions; some drilldown lists by result.

- Reports: Client-side export to XLSX/PDF with minor date normalization.

**11. Configuration**

- Edit connect.phpfor DB credentials.

- If deploying, secure connect.phpoutside web root or guard via server config.

**12. Troubleshooting**

- Blank pages or DB errors: verify credentials in connect.phpand table schema.

- Charts not rendering: ensure Chart.js is loaded (local or CDN) and JSON data is not empty.

- Exports failing: confirm CDN access to SheetJS/jsPDF and correct table selector.

- Session not persisting: check PHP session.save‑path and cookie settings.

**13. Suggested Improvements**

- Replace MD5 with password‑hash/password‑verify and add per-user salts.

- Centralize input validation and use parameterized UPDATEs.

- Add role-based route guards on all sensitive pages.

- Implement audit logging for insert/update actions.

- Add migrations and seeders (SQL files) under sql.

**14. Quick Reference (URLs)**

- Landing: /index.php

- Clerk: /clerk‑login.php ­> /cleark‑dashboard.php

- CO: /co‑login.php ­> /dashboard.php

- Data Entry: /form.html (POST to /insert.php)

- Search/Edit: POST to /search‑soldier.php, then submit to /update‑soldier.php

- Reports: /download‑report.php?type=ppt|bpet|firing|overweight

**End of Document**