Crime Pattern Analysis System (CPAS):

Group Members:

1. Abdullah Irfan 29266
2. Muhammad Fahad 29264
3. Arham Jamshaid 29235

Class No: 99532
DATABASE SYSTEMS
TUE – THU 01:00 pm To 02:15 pm
Maria Rahim

Disclaimer: In accordance with academic integrity, we acknowledge the use of AI-assisted tools during the development and documentation of this project:
The following Large Language Model (LLM) tools were utilized as programming assistants:

- Claude (Sonnet 4.5) by Anthropic
- ChatGPT by OpenAI
- Gemini by Google
- Cursor
- Copilot
- Grok
- Etc

Note: this report has multiple parts please refer to respective pdf wherever mentioned.

Index / Appendix

BUSINESS SCENARIO

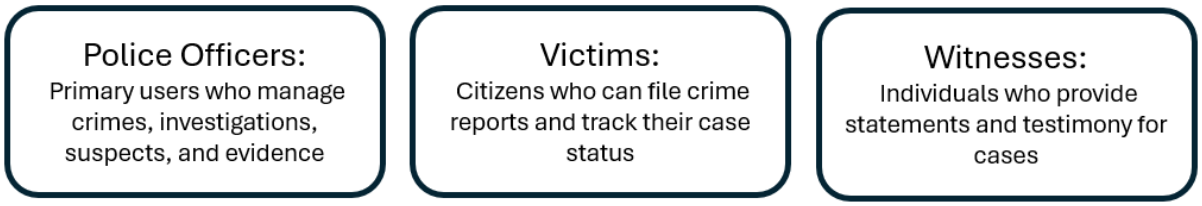
1.1 Description of the Chosen Business Scenario

MehfoozPakistan Crime Pattern Analysis System (CPAS) is a comprehensive law enforcement management system designed to modernize crime tracking, investigation management, and predictive policing in Pakistan. The system addresses the critical need for centralized crime data management, real-time investigation tracking, and data-driven crime prevention strategies.

The system serves as a unified platform for police departments to:

- Record and track criminal incidents with detailed categorization
- Manage investigations from initiation to closure
- Maintain comprehensive suspect, victim, and witness databases
- Coordinate evidence collection and chain of custody
- Generate analytics for crime pattern recognition
- Predict crime hotspots and risk assessments for resource allocation

Target Users:



1.2 Summary of Interview(s) and Application Analysis

Stakeholder Interviews Conducted:

1. SP Abdul Hafeez Junejo (my friend's father):
    - Deficiencies:
        - Manual record-keeping leads to data loss and inconsistencies
        - Difficulty tracking investigation progress across multiple cases
        - No systematic way to identify crime patterns or hotspots
        - Evidence chain of custody often unclear or undocumented

        - Resource allocation inefficient due to lack of predictive analytics
    - Requirements Gathered:
        - Need for centralized database architecture.
        - Real-time dashboard showing active investigations and crime statistics
        - Automated case assignment and officer workload tracking
        - Geographic crime mapping and hotspot identification
        - Predictive tools for proactive policing
2. Crime Victims (relatives that went through street crime) and Average Citizens (Multiple interviews)
    - Deficiencies:
        - Lack of transparency in case progress
        - Difficulty in filing reports (requires physical visit)
        - No mechanism to check case status
        - Fear of retaliation prevents witness cooperation
    - Requirements Gathered:
        - Online crime reporting capability
        - Secure login to track personal case status
        - Confidentiality in witness statements
        - Notification system for case updates

1.3 Application Analysis:

The system was designed with Role-Based Access Control (RBAC) to ensure data security:

- Officer Role: Full access to all crime data, investigations, suspects, evidence, and analytics
- Victim Role: Limited access to personal reports and cases
- Witness Role: Access to cases requiring testimony

Technology Stack Chosen:

- ✓ Frontend: React.js for responsive, single-page application
- ✓ Backend: Node.js with Express for RESTful API
- ✓ Database: Oracle Database for enterprise-grade reliability and advanced features (triggers, stored procedures, views)
- ✓ Authentication: JWT-based secure authentication with bcrypt password hashing

Key Functional Modules Implemented:

1. Crime Management: CRUD operations for crimes with linked suspects, victims, witnesses, and evidence

2. Investigation Tracking: Case assignment, status updates, crime linking, and timeline management
3. Personnel Management: Officers, suspects, victims, and witnesses with complete profiles
4. Evidence Management: Chain of custody tracking with officer accountability
5. Analytics Dashboard: Crime trends, patterns, hotspot identification, and statistics
6. Predictive Analytics: Risk assessment, pattern matching, and crime forecasting using historical data

BUSINESS RULES

Explanation of Core Business Rules and Use Cases

2.1 Crime Recording and Classification

BR-1: Crime Type Categorization

- Every crime must be classified into a predefined category: Violent, Property, Cyber, White-Collar, Drug-Related, or Other
- Each crime must have a specific type (e.g., Murder, Robbery, Fraud) linked to its category
- Prevents ambiguity in crime reporting and ensures consistent analytics

BR-2: Crime Status Workflow

- Valid statuses: Reported → Under Investigation → Solved/Unsolved/Cold Case
- Only officers can update crime status
- Status changes are tracked for audit purposes

BR-3: Location Tracking

- Every crime must be associated with a location (City, Area, Street, Coordinates)
- Duplicate locations are automatically detected and merged
- Enables geographic crime mapping and hotspot analysis

BR-4: Time and Date Validation

- Date_Occurred cannot be in the future (enforced by trigger trg_validate_crime_dates)
- Date_Reported must be on or after Date_Occurred
- System auto-calculates Day_Of_Week from Date_Occurred (via trigger crime_bir)

2.2 Investigation Management

BR-5: Investigation Lifecycle

- Every investigation must have:
  - Unique case number (e.g., INV-2025-001)
  - Lead officer assigned
  - Start date (defaults to current date)
  - Status: Active, Suspended, Cold Case, or Closed

BR-6: Crime-Investigation Linking

- Multiple crimes can be linked to one investigation (serial crimes, related incidents)
- One crime can be part of multiple investigations (complex cases)
- Ensures comprehensive case management

BR-7: Automatic Status Updates

- When all linked crimes are solved, investigation status auto-updates to Closed (via trigger trg_auto_update_investigation_status)
- Reduces manual overhead and ensures data accuracy

BR-8: Investigation Assignment

- Only active officers can be assigned as lead investigators
- Assignment automatically updates investigation status to Active
- Timestamp and officer details appended to investigation notes

2.3 Suspect, Victim, and Witness Management

BR-9: Suspect Data Validation

- Suspects must have: Name, Date of Birth, Physical Description
- Criminal history is optional but tracked
- Trigger trg_validate_suspect_data ensures name is not null and DOB is valid

BR-10: Victim Authentication

- Victims must register with email and password to track their cases
- Passwords hashed using bcrypt for security
- Can only view their own filed reports and linked crimes

BR-11: Witness Statement Management

- Witnesses can be marked as Key Witness for critical cases
- Statement date and text are mandatory
- Witness confidentiality maintained (only officers can view full details)

BR-12: Multi-Entity Crime Linking

- One crime can have multiple suspects, victims, and witnesses
- Bridge tables (Crime_Suspect, Crime_Victim, Crime_Witness) manage relationships
- Prevents data duplication and maintains referential integrity

2.4 Evidence Management

BR-13: Evidence Chain of Custody

- Every evidence item must track:
    - Collected_By (Officer ID)
    - Date_Collected
    - Storage_Location
    - Current status: Collected, In Analysis, Stored

BR-14: Evidence Integrity

- Evidence cannot be deleted if linked to an active investigation (enforced by trigger trg_maintain_evidence_integrity)
- All evidence updates must be performed via chain of custody procedure (sp_update_evidence_chain)
- Actions tracked: COLLECTED, TRANSFERRED, ANALYZED

BR-15: Evidence-Crime Linking

- Evidence must be linked to at least one crime
- Mandatory foreign key ensures orphaned evidence cannot exist
- Supports evidence sharing across related crimes

2.5 Crime Reporting Workflow

BR-16: Citizen Crime Reporting

- Victims or citizens can file reports without creating a crime record immediately
- Reports contain: Victim ID, Reported By Name, Details, Date Reported
- Officers review reports and create formal crime records if valid

BR-17: Automated Report-to-Crime Conversion

- Stored procedure sp_create_crime_report automates:
    1. Creating crime report
    2. Generating crime record from report details
    3. Linking report to crime
    4. All operations in single transaction (atomicity guaranteed)

BR-18: Report Review Process

- Officers can mark reports as Pending Review, Approved, or Rejected
- Only approved reports are converted to crimes
- Prevents spam and false reporting

2.6 Analytics and Predictions

BR-19: Crime Trends Analysis

- Monthly crime trends calculated from historical data
- Grouped by crime type, location, and time period
- View Crime_Trends_Monthly provides pre-aggregated data for performance

BR-20: Crime Pattern Detection

- Patterns identified based on:
    - Day of week (e.g., robberies peak on Fridays)
    - Time of day (e.g., burglaries occur 2-4 AM)
    - Location clusters (e.g., downtown area high-risk zone)
- View Crime_Patterns_Weekly supports pattern analysis

BR-21: Hotspot Identification

- Hotspots ranked by:
    - Total crimes in location
    - Solve rate (solved/total crimes %)
    - Trend (increasing/decreasing)
- View Crime_Hotspots provides location-based aggregates

BR-22: Risk Assessment

- Stored procedure sp_predict_crime_risk calculates risk score (0-100) for given location
- Based on:
    - Historical crime frequency
    - Recent crime trends
    - Seasonal patterns
    - Location characteristics
- Recommend patrol frequency and resource allocation

BR-23: Crime Forecasting

- Predicts future crime counts for next 1-12 months
- Uses linear regression on historical trends
- Helps in budget planning and resource allocation

2.7 Security and Access Control

BR-24: Role-Based Permissions

- Officers: Full CRUD on all entities
- Victims: Read-only on their reports; create new reports
- Witnesses: Read-only on cases they're linked to; update their statements
- Public: No access (must authenticate)

BR-25: Password Security

- Passwords hashed using bcrypt (cost factor 10)
- Minimum 6 characters required
- Never stored in plain text

BR-26: JWT Authentication

- Token expires after 24 hours
- Token contains: User ID, Role, Name
- All protected routes validate token before granting access

BR-27: Audit Logging

- Trigger trg_audit_crime_reports logs all changes to Crime_Report table
- Audit table tracks: User, Action (INSERT/UPDATE/DELETE), Timestamp
- Supports forensic analysis and compliance

2.8 Data Integrity Constraints

BR-28: Referential Integrity

- All foreign keys enforced with ON DELETE CASCADE or ON DELETE SET NULL
- Orphaned records prevented
- Cascade deletes for dependent data (e.g., deleting crime removes linked suspects)

BR-29: Auto-Increment Primary Keys

- All tables use sequences + triggers for auto-generated IDs
- Prevents ID conflicts in concurrent inserts

- Ensures uniqueness across system

BR-30: NOT NULL Constraints

- Critical fields enforced as NOT NULL:
  - Crime: Crime_Type_ID, Date_Occurred, Status
  - Investigation: Case_Number, Start_Date, Status
  - Evidence: Crime_ID, Description, Date_Collected
- Guarantees data completeness

2.9 Core Use Cases

UC-1: Officer Records New Crime

1. Officer logs in with credentials
2. Navigates to "Crimes" → "Add New Crime"
3. Selects crime type, location, date/time, status
4. Optionally links suspects, victims, witnesses, evidence
5. System validates data and inserts into database
6. Auto-generates Crime ID and Day of Week
7. Returns success confirmation

UC-2: Citizen Files Crime Report

1. Victim/citizen logs in or creates account
2. Navigates to "File Report"
3. Enters crime details, location, date
4. System creates crime report record
5. Officer reviews report in "Reports" section
6. Officer converts report to formal crime (calls sp_create_crime_report)
7. System auto-links report to crime

UC-3: Officer Creates Investigation

1. Officer creates new investigation with case number
2. Assigns lead officer (self or colleague)
3. Links related crimes to investigation
4. System auto-updates status to "Active"
5. Officer adds notes and updates as case progresses
6. When all crimes solved, system auto-closes investigation

UC-4: Officer Updates Evidence Chain

1. Officer collects evidence at crime scene
2. Records evidence details (description, location, type)
3. Calls "Update Chain of Custody"
4. Selects action: COLLECTED / TRANSFERRED / ANALYZED
5. System updates evidence record via sp_update_evidence_chain

6. Logs officer ID, timestamp, and action notes

UC-5: Analytics - Crime Hotspot Identification

1. Officer navigates to "Analytics" → "Crime Hotspots"
2. System queries Crime_Hotspots view
3. Displays top 10 locations by crime count
4. Shows solve rate and trend for each location
5. Officer uses data to allocate patrol resources

UC-6: Predictive Policing - Risk Assessment

1. Officer navigates to "Predictions" → "Risk Assessment"
2. Enters location (city, area)
3. System calls sp_predict_crime_risk
4. Returns risk score (0-100) and recommendations
5. Officer plans preventive patrols based on risk

Entities, Attributes, and Relationships

3.1 Detailed description of entities, attributes, and relationships including multiplicity constraints.

3.11 Core Entities and Attributes

Core Entities	Bridge Entities
Crime_Type	Crime_Suspect
Location	Crime_Victim
Officer	Crime_Witness
Suspect	Investigation_Crime
Victim	Report_Crime
Witness	
Investigation	
Crime	
Evidence	
Crime_Report	

1. Crime_Type

- Crime_Type_ID (Primary Key)
- Type_Name
- Category
- Description

2. Location

- Location_ID (Primary Key)
- District
- City
- Province
- Latitude
- Longitude

3. Officer

- Officer_ID (Primary Key)
- Name
- Contact_No
- Email
- Password

4. Suspect

- Suspect_ID (Primary Key)
- Name
- Age
- Gender
- Contact_No
- Address
- Criminal_Record

5. Victim

- Victim_ID (Primary Key)
- Name
- Age
- Gender
- Contact_No
- Address
- Email
- Password

6. Witness

- Witness_ID (Primary Key)
- Name
- Age
- Gender
- Contact_No
- Address
- Email
- Password

7. Investigation

- Investigation_ID (Primary Key)
- Investigation_Name

- Start_Date
- End_Date
- Status
- Outcome
- Lead_Officer_ID (Foreign Key → Officer)

8. Crime

- Crime_ID (Primary Key)
- Date_Reported
- Date_Occurred
- Time_Occurred
- Day_of_Week
- Status
- Severity_Level
- Description
- Crime_Type_ID (Foreign Key → Crime_Type)
- Location_ID (Foreign Key → Location)

9. Evidence

- Evidence_ID (Primary Key)
- Type
- Description
- Date_Collected
- Collected_By (Foreign Key → Officer)
- Crime_ID (Foreign Key → Crime)

10. Crime_Report

- Report_ID (Primary Key)
- Report_Date
- Report_Description
- Report_Status
- Filed_By_Victim_ID (Foreign Key → Victim)
- Filed_By_Witness_ID (Foreign Key → Witness)

3.12 Relationships and Multiplicity

One-to-Many (1:N) Relationships:

1. Crime_Type → Crime
    - One crime type can be associated with many crimes
    - Each crime belongs to exactly one crime type
2. Location → Crime
    - One location can have many crimes
    - Each crime occurs at exactly one location
3. Officer → Investigation (Lead Officer)
    - One officer can lead many investigations
    - Each investigation has exactly one lead officer
4. Officer → Evidence (Collected By)
    - One officer can collect many pieces of evidence
    - Each evidence is collected by exactly one officer
5. Crime → Evidence
    - One crime can have many pieces of evidence
    - Each evidence belongs to exactly one crime
6. Victim → Crime_Report (Filed By)
    - One victim can file many crime reports
    - Each report is filed by exactly one victim OR one witness (mutually exclusive)
7. Witness → Crime_Report (Filed By)
    - One witness can file many crime reports
    - Each report is filed by exactly one victim OR one witness (mutually exclusive)

Many-to-Many (M:N) Relationships:

1. Crime ↔ Suspect (via Crime_Suspect)
    - One crime can involve many suspects
    - One suspect can be involved in many crimes
2. Crime ↔ Victim (via Crime_Victim)
    - One crime can have many victims
    - One victim can be affected by many crimes
3. Crime ↔ Witness (via Crime_Witness)
    - One crime can have many witnesses
    - One witness can witness many crimes
4. Investigation ↔ Crime (via Investigation_Crime)
    - One investigation can cover many crimes
    - One crime can be part of many investigations
5. Crime_Report ↔ Crime (via Report_Crime)
    - One crime report can reference many crimes

- One crime can be referenced in many reports

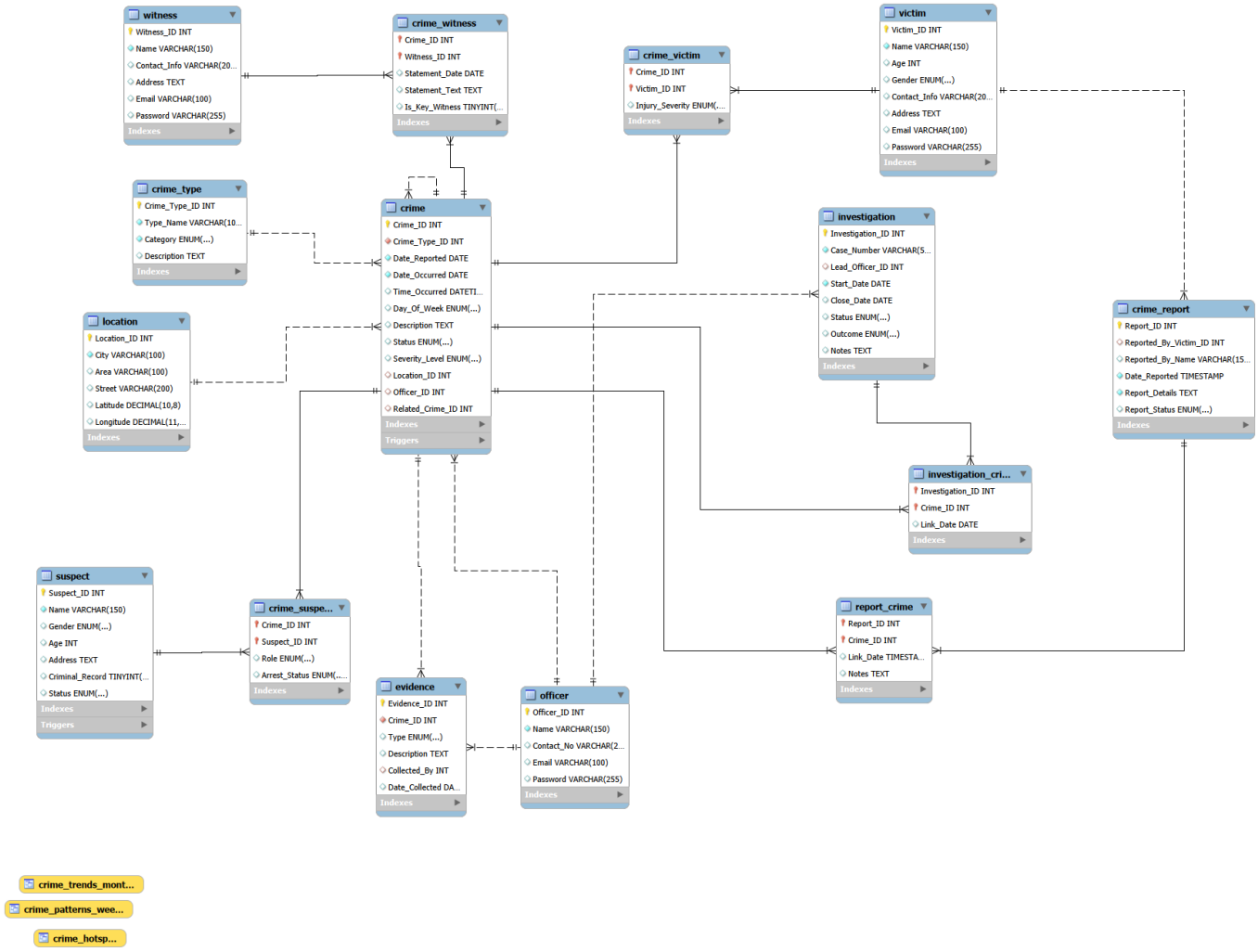
Referential Integrity Constraints

The system maintains referential integrity through 18 foreign key constraints:

1. Crime.Crime_Type_ID references Crime_Type.Crime_Type_ID
2. Crime.Location_ID references Location.Location_ID
3. Investigation.Lead_Officer_ID references Officer.Officer_ID
4. Evidence.Collected_By references Officer.Officer_ID
5. Evidence.Crime_ID references Crime.Crime_ID
6. Crime_Report.Filed_By_Victim_ID references Victim.Victim_ID
7. Crime_Report.Filed_By_Witness_ID references Witness.Witness_ID
8. Crime_Suspect.Crime_ID references Crime.Crime_ID
9. Crime_Suspect.Suspect_ID references Suspect.Suspect_ID
10. Crime_Victim.Crime_ID references Crime.Crime_ID
11. Crime_Victim.Victim_ID references Victim.Victim_ID
12. Crime_Witness.Crime_ID references Crime.Crime_ID
13. Crime_Witness.Witness_ID references Witness.Witness_ID
14. Investigation_Crime.Investigation_ID references Investigation.Investigation_ID
15. Investigation_Crime.Crime_ID references Crime.Crime_ID
16. Report_Crime.Report_ID references Crime_Report.Report_ID
17. Report_Crime.Crime_ID references Crime.Crime_ID

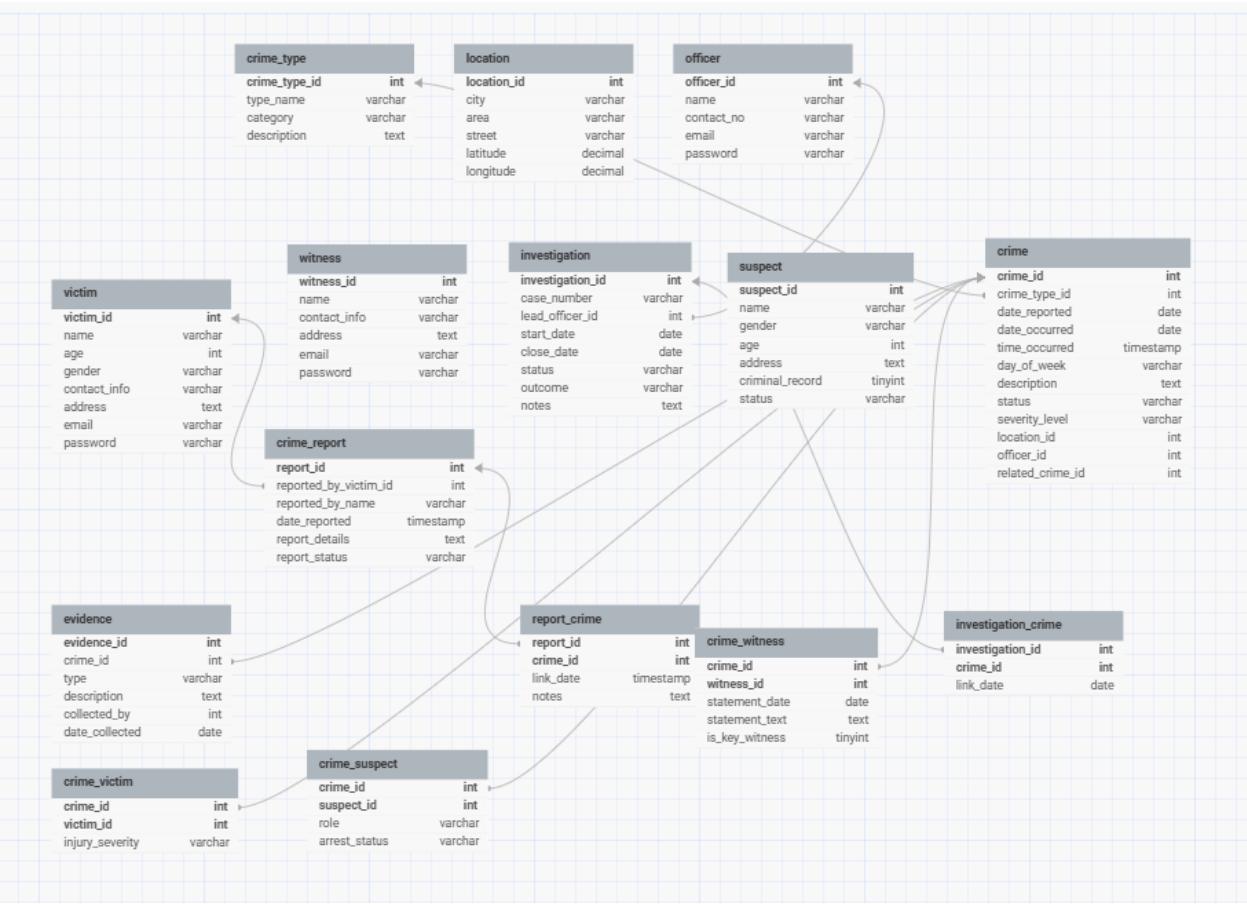
All foreign keys enforce CASCADE ON DELETE for Crime relationships (automatic cleanup when a crime is deleted) and SET NULL for optional relationships like Crime_Report filed-by references (preserving historical reports even if user accounts are deleted).

3.2 ER diagram



Relational Schema

4.1 A visual representation of your database schema (using DBDesigner)



4.2 Show and validate normalization steps up to 3rd Normal Form (3NF)

Initial Unnormalized Form (UNF)

Before normalization, crime data might have been stored in a single flat table with repeating groups:
Crime_Record (UNF):
Crime_ID, Date_Reported, Date_Occurred, Time_Occurred, Status, Severity_Level, Description, Crime_Type_Name, Crime_Category, Crime_Description, District, City, Province, Latitude, Longitude, {Suspect_Name, Suspect_Age, Suspect_Gender, Suspect_Contact, Suspect_Address, Suspect_Criminal_Record}, {Victim_Name, Victim_Age, Victim_Gender, Victim_Contact, Victim_Address, Victim_Email}, {Witness_Name, Witness_Age, Witness_Gender, Witness_Contact, Witness_Address, Witness_Email}, {Evidence_Type, Evidence_Description, Date_Collected, Collector_Name, Collector_Contact}, Investigation_Name, Start_Date, End_Date, Investigation_Status, Outcome, Lead_Officer_Name, Lead_Officer_Contact, Lead_Officer_Email, Report_Date, Report_Description, Report_Status

Problems with UNF:

- Repeating groups for suspects, victims, witnesses, and evidence
- Multiple values in single attributes
- Data redundancy (crime type details, location details repeated for each crime)
- Update anomalies (changing officer contact requires updating multiple crime records)
- Insertion anomalies (cannot add a crime type without having a crime)
- Deletion anomalies (deleting last crime of a type loses crime type information)

First Normal Form (1NF)

Crime (1NF):
Crime_ID (PK), Date_Reported, Date_Occurred, Time_Occurred, Day_of_Week, Status, Severity_Level, Description, Crime_Type_Name, Crime_Category, Crime_Type_Description, District, City, Province, Latitude, Longitude

Crime_Suspect (1NF):
Crime_ID (PK), Suspect_Name (PK), Suspect_Age, Suspect_Gender, Suspect_Contact, Suspect_Address, Suspect_Criminal_Record
Crime_Victim (1NF):
Crime_ID (PK), Victim_Name (PK), Victim_Age, Victim_Gender, Victim_Contact, Victim_Address, Victim_Email, Victim_Password

Crime_Witness (1NF):
Crime_ID (PK), Witness_Name (PK), Witness_Age, Witness_Gender, Witness_Contact, Witness_Address, Witness_Email, Witness_Password

Evidence (1NF):
Evidence_ID (PK), Crime_ID, Type, Description, Date_Collected, Collector_Name, Collector_Contact, Collector_Email

Investigation (1NF):
Investigation_ID (PK), Crime_ID, Investigation_Name, Start_Date, End_Date, Status, Outcome, Lead_Officer_Name, Lead_Officer_Contact, Lead_Officer_Email

Crime_Report (1NF):
Report_ID (PK), Crime_ID, Report_Date, Report_Description, Report_Status, Filed_By_Name, Filed_By_Type, Filed_By_Contact

Achievement:

- All attributes now contain atomic values
- Each cell contains a single value
- No repeating groups
- Each row is uniquely identifiable

Second Normal Form (2NF)

Crime_Suspect (1NF):

- Composite Key: (Crime_ID, Suspect_Name)
- Partial Dependencies:
    - Suspect_Age, Suspect_Gender, Suspect_Contact, Suspect_Address, Suspect_Criminal_Record depend only on Suspect_Name, not on Crime_ID
    - This violates 2NF

Transformation to 2NF:
Suspect (2NF):
Suspect_ID (PK), Name, Age, Gender, Contact_No, Address, Criminal_Record

Crime_Suspect (2NF):
Crime_ID (PK, FK), Suspect_ID (PK, FK)

Victim (2NF):
Victim_ID (PK), Name, Age, Gender, Contact_No, Address, Email, Password

Crime_Victim (2NF):
Crime_ID (PK, FK), Victim_ID (PK, FK)

Witness (2NF):
Witness_ID (PK), Name, Age, Gender, Contact_No, Address, Email, Password

Crime_Witness (2NF):
Crime_ID (PK, FK), Witness_ID (PK, FK)

Crime_Type (2NF):
Crime_Type_ID (PK), Type_Name, Category, Description

Location (2NF):
Location_ID (PK), District, City, Province, Latitude, Longitude

Officer (2NF):
Officer_ID (PK), Name, Contact_No, Email, Password
Crime (2NF):
Crime_ID (PK), Date_Reported, Date_Occurred, Time_Occurred, Day_of_Week, Status, Severity_Level, Description, Crime_Type_ID (FK), Location_ID (FK)

Evidence (2NF):
Evidence_ID (PK), Type, Description, Date_Collected, Collected_By (FK to Officer), Crime_ID (FK)

Investigation (2NF):
Investigation_ID (PK), Investigation_Name, Start_Date, End_Date, Status, Outcome, Lead_Officer_ID (FK)

Investigation_Crime (2NF):
Investigation_ID (PK, FK), Crime_ID (PK, FK)

Crime_Report (2NF):
Report_ID (PK), Report_Date, Report_Description, Report_Status, Filed_By_Victim_ID (FK), Filed_By_Witness_ID (FK)

Report_Crime (2NF):
Report_ID (PK, FK), Crime_ID (PK, FK)

Achievement:

- All partial dependencies eliminated
- Non-key attributes fully depend on entire primary key
- Reduced data redundancy significantly

Third Normal Form (3NF)

Transitive Dependency Analysis:
Examining each 2NF table for transitive dependencies:
Crime (2NF):

- Crime_ID → Crime_Type_ID → Type_Name, Category, Crime_Type_Description (transitive dependency - ALREADY REMOVED in 2NF)
- Crime_ID → Location_ID → District, City, Province, Latitude, Longitude (transitive dependency - ALREADY REMOVED in 2NF)
- No remaining transitive dependencies

Location (2NF):

- Location_ID → Province → (potential: Province_Region, Province_Code)
- However, in our schema, Province is a simple attribute with no further dependencies
- No transitive dependencies

Transitive Dependency Analysis:
Examining each 2NF table for transitive dependencies:

Crime (2NF):

- Crime_ID → Crime_Type_ID → Type_Name, Category, Crime_Type_Description (transitive dependency - ALREADY REMOVED in 2NF)
- Crime_ID → Location_ID → District, City, Province, Latitude, Longitude (transitive dependency - ALREADY REMOVED in 2NF)
- No remaining transitive dependencies

Location (2NF):

- Location_ID → Province → (potential: Province_Region, Province_Code)
- However, in our schema, Province is a simple attribute with no further dependencies

- No transitive dependencies

Crime_Report (2NF):

- Report_ID → Filed_By_Victim_ID → Victim details (transitive dependency - ALREADY REMOVED by using FK)
- Report_ID → Filed_By_Witness_ID → Witness details (transitive dependency - ALREADY REMOVED by using FK)
- No remaining transitive dependencies

Final 3NF Schema:

All 15 tables in the current schema are in 3NF:

Core Entities (3NF):

1. Crime_Type(Crime_Type_ID, Type_Name, Category, Description)
2. Location(Location_ID, District, City, Province, Latitude, Longitude)
3. Officer(Officer_ID, Name, Contact_No, Email, Password)
4. Suspect(Suspect_ID, Name, Age, Gender, Contact_No, Address, Criminal_Record)
5. Victim(Victim_ID, Name, Age, Gender, Contact_No, Address, Email, Password)
6. Witness(Witness_ID, Name, Age, Gender, Contact_No, Address, Email, Password)
7. Investigation(Investigation_ID, Investigation_Name, Start_Date, End_Date, Status, Outcome, Lead_Officer_ID)
8. Crime(Crime_ID, Date_Reported, Date_Occurred, Time_Occurred, Day_of_Week, Status, Severity_Level, Description, Crime_Type_ID, Location_ID)
9. Evidence(Evidence_ID, Type, Description, Date_Collected, Collected_By, Crime_ID)
10. Crime_Report(Report_ID, Report_Date, Report_Description, Report_Status, Filed_By_Victim_ID, Filed_By_Witness_ID)

Bridge Entities (3NF):

11. Crime_Suspect(Crime_ID, Suspect_ID)
12. Crime_Victim(Crime_ID, Victim_ID)
13. Crime_Witness(Crime_ID, Witness_ID)
14. Investigation_Crime(Investigation_ID, Crime_ID)
15. Report_Crime(Report_ID, Crime_ID)

Achievement:

- All transitive dependencies eliminated
- Each non-key attribute depends directly on the primary key
- Minimal data redundancy
- Maximum data integrity
- Optimal structure for CRUD operations

Verification of 3NF Compliance

Functional Dependencies (Sample):
Crime_Type:

- Crime_Type_ID → Type_Name, Category, Description

Location:

- Location_ID → District, City, Province, Latitude, Longitude

Officer:

- Officer_ID → Name, Contact_No, Email, Password

Crime:

- Crime_ID → Date_Reported, Date_Occurred, Time_Occurred, Day_of_Week, Status, Severity_Level, Description, Crime_Type_ID, Location_ID

Investigation:

- Investigation_ID → Investigation_Name, Start_Date, End_Date, Status, Outcome, Lead_Officer_ID

Evidence:

- Evidence_ID → Type, Description, Date_Collected, Collected_By, Crime_ID

All functional dependencies follow the pattern: Primary_Key → Non-Key_Attributes, with no transitive or partial dependencies.

Conclusion:
The MehfoozPakistan Crime Pattern Analysis System database is fully normalized to Third Normal Form (3NF), ensuring data integrity, minimal redundancy, and optimal performance for law enforcement operations.

4.3 DDL script screenshots or text snippets

Database Schema Overview

The database schema implements a fully normalized design with the following components:

a. 10 Core Tables: Crime Type, Location, Officer, Suspect, Victim, Witness, Investigation, Crime, Evidence, Crime Report
b. 5 Bridge Tables: Crime Suspect, Crime Victim, Crime Witness, Investigation Crime, Report Crime
c. 10 Sequences: For auto-increment functionality • 10 Auto-Increment Triggers: For primary key generation
d. 7 Business Logic Triggers: For validation and automation
e. 5 Stored Procedures: For complex operations
f. Analytical Views: For crime trends and patterns

4.31 Explain the constraints applied to your tables. +
4.32 Explain any triggers, stored procedures, and views that you have created.

Please double click the following icon to redirect to the page explaining the DDL script

DDL script
screenshots or text sn

4.33 Show the insertion of some data that you add to test your application

Sample data for testing the MehfoozPakistan CPAS application was generated using a Python script that leveraged the library for realistic Pakistani data generation and for database connectivity.

```
C:\Users\Dell>python "d:\db backup\FINAL prev\populate_database.py"
================================================================================
   CPAS - Crime Pattern Analysis System
   Enhanced Realistic Oracle Database Data Generator
================================================================================
```

```
================================================================
 ENHANCED DATA GENERATION COMPLETED SUCCESSFULLY!
================================================================

 Summary Statistics:
   • Crime Types: 20 (Violent, Property, Cyber, Drug-Related, White-Collar)
   • Locations: 42 (Mix of High/Medium/Low crime areas)
   • Officers: 20 (Inspectors, SIs, ASIs)
   • Suspects: 80 (15 repeat offenders, 45 first-time, 20 unknown)
   • Victims: 100
   • Witnesses: 70
   • Crimes: 281 (With realistic time/location patterns)
   • Crime-Suspect Links: 406
   • Crime-Victim Links: 324
   • Crime-Witness Links: 157
   • Evidence Items: 358 (Crime-specific evidence types)
   • Investigations: 50 (Active, Closed, Suspended, Cold Case)
   • Investigation-Crime Links: 81
   • Crime Reports: 100 (From victims and citizens)
   • Report-Crime Links: 87

 Database is ready with realistic, interpretable data!
 Data includes patterns for temporal, spatial, and categorical analysis
 Mix of solved/unsolved cases, repeat/first-time offenders
 Realistic injury severities, evidence types, and witness statements

 Database connection closed.
```

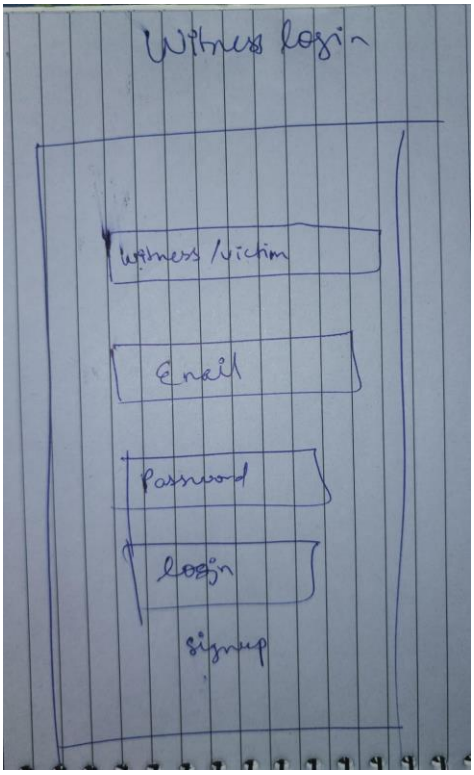
Application Flow

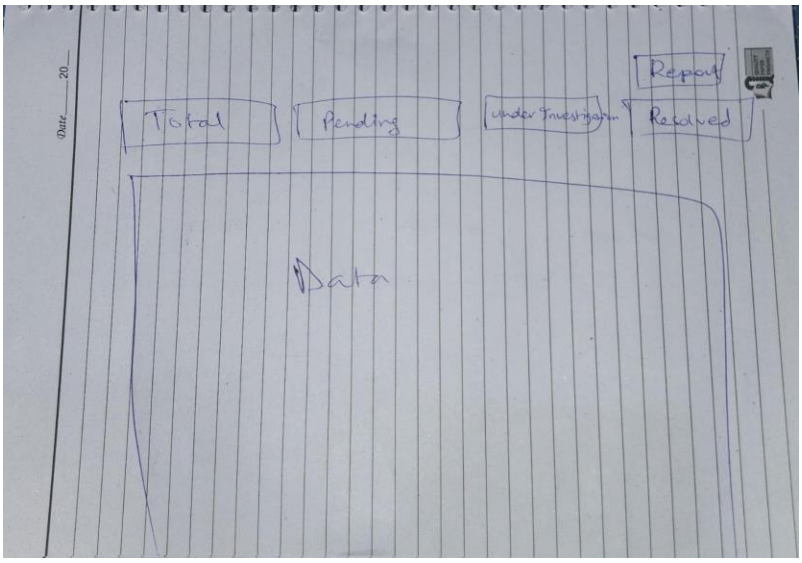
5.1 Flow diagram showing how users interact with your application.

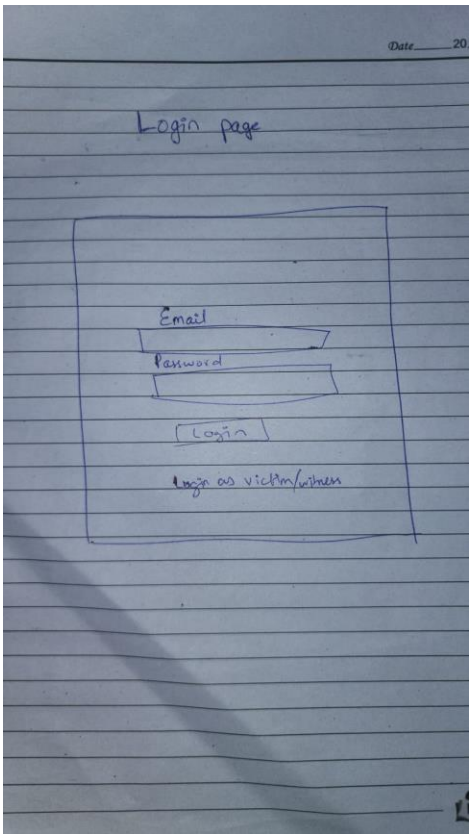
Please refer to the **Flow diagram showing how users interact with your application pdf** if below link does not work

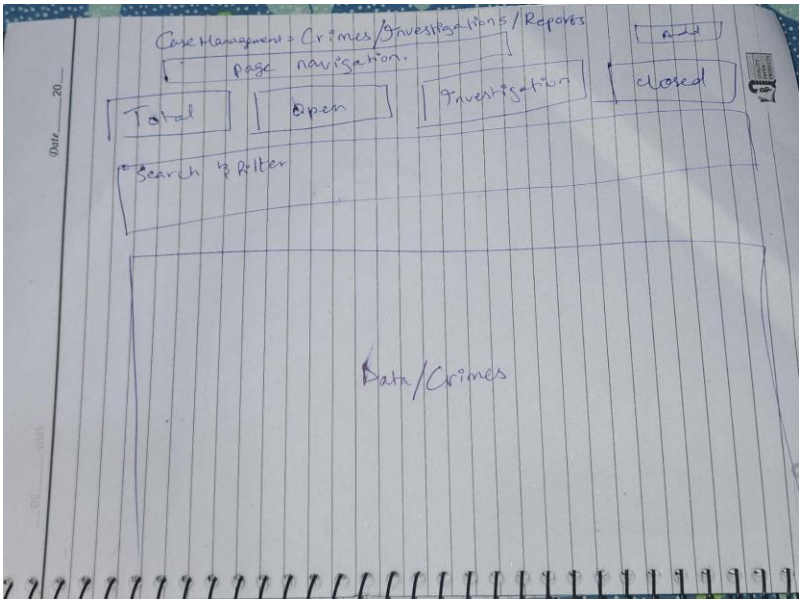
Flow diagram showing how users in

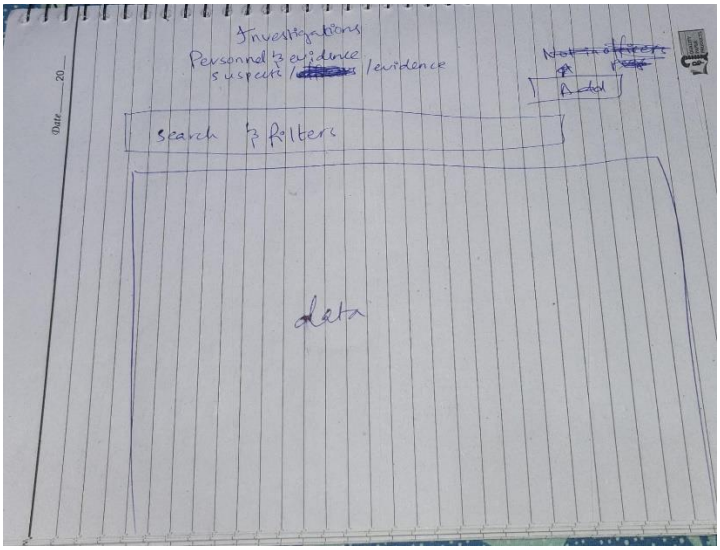
5.2 Wireframes or sketches of your application's UI

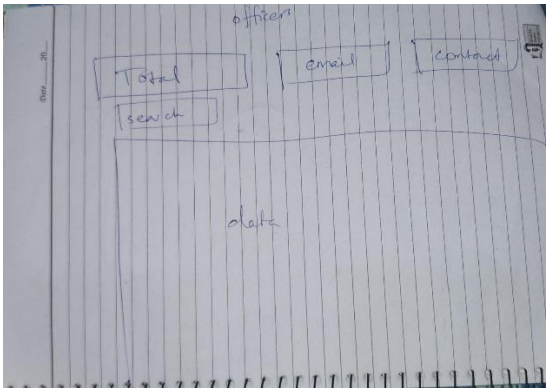


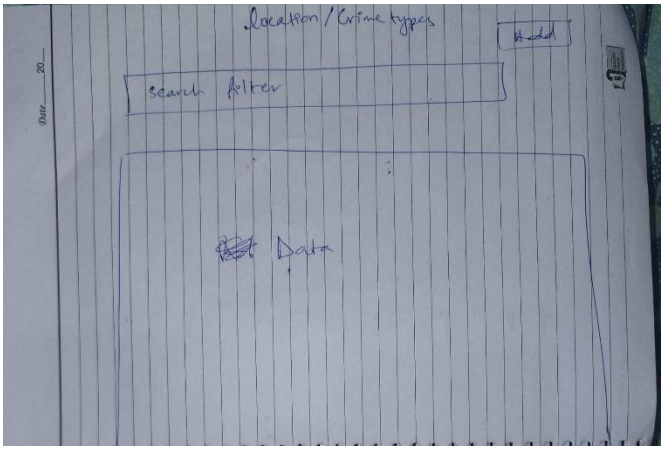

Date 20
Report
Total
Pending
under Investigation
Resolved
Data

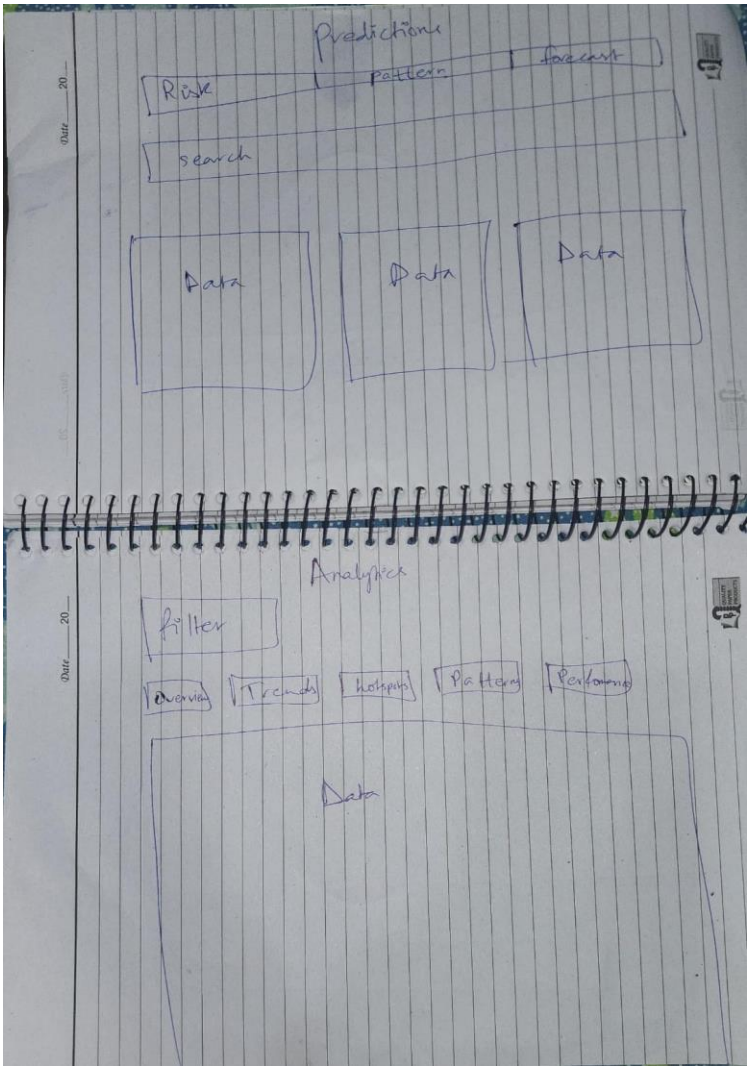

Date 20
Login page
Email
Password
Login
Login as victim/witness


Case Management - Crimes/Investigations/Reports
Page navigation.
Add
Total
Open
Investigation
closed
Search & Filter
Data/Crimes


Investigations
Personnel & evidence
suspects / evidence
Add
Search & Filters
data


officers
Total
email
contact
search
data


location / Crime types
search filter
Data


Predictions
Risk
pattern
forecast
search
Data
Data
Data
Analytics
filter
Data

Page-by-Page Navigation and SQL Queries

Okay this may sound weird but out app has too many functionalities hence I have attached a video exploring each page in the application and the document below explains each SQL query. (Double click the icons below)

Document explaining my sql querie(ignore places where snapshots are to be uploaded instead refer to the videos below)

Please refer to **Page-by-Page Navigation and SQL Queries pdf** if below link does not work

Page-by-Page Navigation and SQL C

Videos which explore the entire application:

Please refer to google rive which has our application walkthrough video if below links do not work.

Officer side.mkv

Victim and Witness side.mkv

Work Contribution

We, **Abdullah Irfan (29266)**, **Muhammad Fahd (29264)**, and **Arham Jamshaid**, confirm that every part of the *MehfoozPakistan – Crime Pattern Analysis System (CPAS)* was planned, designed, developed, tested, and written by us together as a team.

We Confirm That:
Most of the work was done through pair or group programming
All of us took part in the database, backend, frontend, analytics, and testing
No important task was done by any one person alone
Every piece of code was reviewed and approved by the team
All documents were created and finalized together
Each member understands the entire system from start to finish
The workload throughout the project was shared fairly and evenly

Equal Contribution
We all agree that the contribution was equal in terms of time, effort, technical work, and decision-making.