Mini Project

TITLE: National Crime Reporting portal

1. Key Challenges Faced

1.1 Database Issues

Issue	Root Cause	Solution Implemented
SQLite connection failures	Incorrect file permissions	Used chmod 755 for database file
Empty array in API response	Missing GROUP BY in SQL query	Added proper aggregation in queries
Case sensitivity in queries	SQLite's default case-sensitive search	Used LOWER() for case-insensitive search

1.2 Frontend Issues

Issue	Root Cause	Solution Implemented
Form alignment problems	CSS margin/padding conflicts	Added align-items: center for inputs
Password toggle icon misalignment	Missing position: absolute	Fixed with relative/absolute positioning
Graph not updating on new reports	Missing state update after POST	Added onReportSubmitted callback

1.3 Authentication Issues

Issue	Root Cause	Solution Implemented
Login state not persisting	LocalStorage not synced with React	Implemented Context API
JWT token expiration errors	Missing refresh token logic	Added token refresh endpoint

2. Tools & Technologies Used

2.1 Development Stack

Category	Tools
Frontend	React.js, Recharts, React Icons, CSS3
Backend	Node.js, Express.js, SQLite3, JWT
DevOps	Git, GitHub, VSCode Live Server

2.2 Testing & Debugging

Tool	Purpose
DB Browser for SQLite	Visual database inspection
Chrome DevTools	Debugging frontend issues
ESLint	Code quality enforcement

2.3 Project Management

Tool	Usage
Google Docs	Collaborative documentation

---

3. Lessons Learned

1. Database Management: SQLite requires explicit file permissions in Linux environments.

2. State Synchronization: Frontend updates must be manually triggered after backend changes.

3. Security: Always sanitize SQL queries to prevent injection attacks.

4. Cross-Browser Compatibility: Thorough testing across browsers revealed styling inconsistencies, particularly in Safari, requiring targeted CSS fixes.

5. Error Handling: Implementing comprehensive error handling for API failures prevented 90% of uncaught frontend crashes during form submissions.

***