## Analysis of Query Management Enhancements

### 1. Enhanced Query Management Features

#### Advanced Search

- **Description**: Allow users to search for any query without relying on dropdown menus.
- **Changes Needed**:
    - Replace the dropdown menu for query selection with a text-based advanced search input field.
    - Implement backend search functionality to handle full-text search or specific query fields (e.g., query ID, keywords, etc.).
    - Update the front-end to include a real-time search feature with suggestions or filters.
- **Feasibility**: High. Modern frameworks like asp.net can easily integrate search functionalities, and the back-end can use database query capabilities (like SQL LIKE or Elasticsearch).

#### Department Dropdown

- **Description**: Add a dropdown menu to filter queries by department for targeted management.
- **Changes Needed**:
    - Add a department field to the queries table in the database, if not already present.
    - Update the front-end UI to include a dropdown for department selection.
    - Update the back-end API to accept a department filter and return results accordingly.
- **Feasibility**: High. Dropdown filtering is a standard feature and straightforward to implement.

### 2. File Management Enhancements

#### Multiple File Uploads

- **Description**: Enable users to attach multiple files using drag-and-drop functionality.

- **Changes Needed**:
    - Update the file upload UI to support multiple file selection and drag-and-drop.
    - Modify the back-end to accept and store multiple files per query (e.g., update the database schema to associate multiple files with a single query).
    - Implement file validation (e.g., file type, size limits).
- **Feasibility**: High. Libraries like react-drop zone (or native HTML5 drag-and-drop) can be used for the front-end, and backend storage solutions like AWS S3 or local storage can handle multiple uploads.

**File Preview**

- **Description**: Allow users to preview uploaded files directly in the system.
- **Changes Needed**:
    - Add file preview functionality for common file types (e.g., images, PDFs, text files).
    - Implement a viewer component in the front-end.
    - Ensure secure file access (e.g., token-based access URLs for file previews).
- **Feasibility**: Medium-High. File previews are straightforward for common file types but may require additional steps for complex formats (e.g., MS Office documents).

---

### 3. Notification System

**Email Alerts**

- **Description**: Notify users via email about new assignments or pending queries.
- **Changes Needed**:
    - Set up email notifications using a library or service like NodeMailer, SendGrid, or AWS SES.
    - Update the backend to trigger email notifications for specific events (e.g., query assignment, query nearing SLA deadlines).
    - Add email templates for each notification type.

- **Feasibility**: High. Email notifications are standard and supported by many libraries.

### Status Updates

- **Description**: Send automated updates to clients on query progress or resolution.
- **Changes Needed**:
  - Add automated status tracking in the backend (e.g., query status changes triggering an update).
  - Integrate with the email notification system for client updates.
  - Allow clients to opt-in/opt-out of updates.
- **Feasibility**: High. This builds on the existing email notification system.

---

## 4. Support Enhancements

### Help Centre Integration

- **Description**: Embed a help centre or FAQ section for quick assistance.
- **Changes Needed**:
  - Create a Help Centre page or section in the front-end UI.
  - Add a database or static file storage for FAQ content.
  - Optionally, integrate a search function for FAQs.
- **Feasibility**: High. Embedding FAQs is simple, and search functionality is optional but enhances usability.