

# JAVA BACKEND PROJECT REPORT

## Project: CRUD Operations with Search (Select by Date) Feature

Name: Divine AKISA

Date: 12/02/2026

Course Name: Backend Using Java

### 1. Introduction

This report presents the implementation of CRUD (Create, Read, Update, Delete) operations combined with a Search (Select by Date) feature in a Java backend application using JDBC and a relational database. The project demonstrates how database records can be managed efficiently and retrieved based on a specific date field.

### 2. Project Overview

The system manages records stored in a relational database table. For demonstration purposes, the project example used is a child health monitor system.

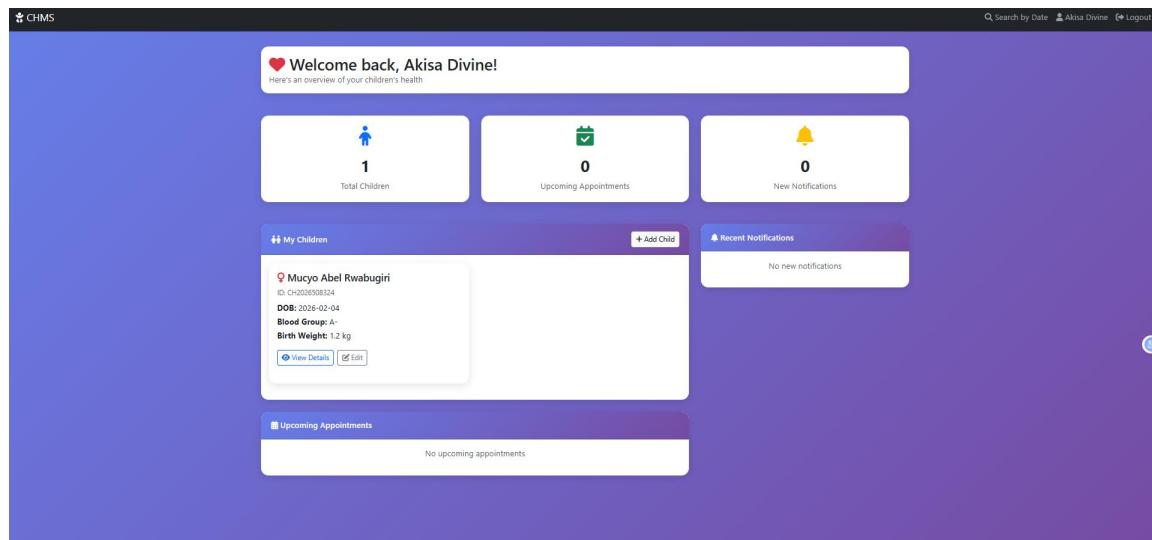
The appointment\_date field was selected for the Search feature because it allows retrieval of transactions performed on a specific date, which is a common real-world requirement in financial systems.

### 3. CRUD Operations Implementation

#### 3.1 Create Operation

The Create operation inserts new records into the database using an SQL INSERT statement. PreparedStatement is used to safely pass parameters to prevent SQL injection.

BeforeInsert Second Children



SQl Server: 127.0.0.1 + Database: chms\_db + table: children

Showing rows 0 - 8 (9 total). Query took 0.0012 seconds.

SELECT \* FROM `children`

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 | Filter rows | Search this table | Sort by key: None | Extra options

	child_id	unique_profile_id	full_name	date_of_birth	gender	birth_weight	birth_height	blood_group	mother_id	father_name	father_phone	emergency_contact	address	medical_history	is_active	created_at	updated_at
<input type="checkbox"/>	1	CH0202301024	Emma Wilson	2023-03-15	FEMALE	3.20	50.00	O+	5	Robert Wilson	+12345678909	+12345678909	123 Maple Street, Springfield, IL, 62701	No known allergies	1	2020-02-12 11:21:42	2020-02-12 11:21:42
<input type="checkbox"/>	2	CH0202400245	Oliver Wilson	2024-06-20	MALE	3.50	52.00	O+	5	Robert Wilson	+12345678909	+12345678909	123 Maple Street, Springfield, IL, 62701	No known allergies	1	2020-02-12 11:21:42	2020-02-12 11:21:42
<input type="checkbox"/>	3	CH02023030458	Sophia Brown	2023-01-10	FEMALE	3.00	49.00	A+	6	James Brown	+1234567810	+1234567810	456 Oak Avenue, Chicago, IL, 60614	Lactose intolerance	1	2020-02-12 11:21:42	2020-02-12 11:21:42
<input type="checkbox"/>	4	CH02024004567	Noah Brown	2024-05-05	MALE	3.00	53.00	A+	6	James Brown	+1234567810	+1234567810	456 Oak Avenue, Chicago, IL, 60614	No known allergies	1	2020-02-12 11:21:42	2020-02-12 11:21:42
<input type="checkbox"/>	5	CH02020905787	Isabella Davis	2022-11-22	FEMALE	3.30	51.00	B+	7	William Davis	+1234567811	+1234567811	789 Pine Road, Peoria, IL, 61602	Peanut allergy	1	2020-02-12 11:21:42	2020-02-12 11:21:42
<input type="checkbox"/>	6	CH02025005789	Liam Davis	2025-01-12	MALE	3.60	51.50	B+	7	William Davis	+1234567811	+1234567811	789 Pine Road, Peoria, IL, 61602	No known allergies	1	2020-02-12 11:21:42	2020-02-12 11:21:42
<input type="checkbox"/>	7	CH0202307390	Ava Miller	2023-07-08	FEMALE	3.10	49.50	AB+	8	Richard Miller	+1234567812	+1234567812	321 Elm Street, Rockford, IL, 61101	Eczema	1	2020-02-12 11:21:42	2020-02-12 11:21:42
<input type="checkbox"/>	8	CH02024008901	Ethan Moore	2024-02-14	MALE	3.40	51.00	O-	9	Charles Moore	+1234567813	+1234567813	654 Birch Lane, Naperville, IL, 60540	No known allergies	1	2020-02-12 11:21:42	2020-02-12 11:21:42
<input type="checkbox"/>	9	CH0202500324	Muciyo Abiel Rivabugn	2020-02-04	FEMALE	1.20	0.50	A-	10	Manzi	0788888884	0788888884	Kicukiro	None	1	2020-02-12 13:00:36	2020-02-12 15:59:30

Show all | Number of rows: 25 | Filter rows | Search this table | Sort by key: None | Extra options

Query results operations

Print | Copy to clipboard | Export | Display chart | Create view

Bookmark this SQL query

Label:  Let every user access this bookmark

Bookmark this SQL query

## After Insertion

Add New Child

Child's Full Name \*: Muriyana ANGE Date of Birth \*: 02/06/2026

Gender \*: Female Blood Group: B+

Birth Weight (kg) \*: 1.2 Birth Height (cm) \*: 0.3

Father's Name: mUKIZA Father's Phone: 0788888884

Emergency Contact: 0788864532

Address: Kicukiro

Medical History / Allergies: E.g., No known allergies, Asthma, Peanut allergy, etc.

[← Back to Dashboard](#) [Add Child](#)

**Welcome back, Akisa Divine!**  
Here's an overview of your children's health

**Total Children:** 2

**Upcoming Appointments:** 0

**New Notifications:** 0

**Recent Notifications:** No new notifications

Showing rows 0 - 9 (10 total). Query took 0.0008 seconds.

SELECT \* FROM `children`

Profile | Edit inline | Explain SQL | Create PHP code | Refresh |

Show all | Number of rows: 25 | Filter rows | Search this table | Sort by key: None

Extra options:  Check all |  With selected |  Edit |  Copy |  Delete |  Export

	child_id	unique_profile_id	full_name	date_of_birth	gender	birth_weight	birth_height	blood_group	mother_id	father_name	father_phone	emergency_contact	address	medical_history	is_active	created_at	updated_at
<input type="checkbox"/>	1	CH2023001234	Emma Wilson	2023-03-15	FEMALE	3.20	50.00	0+	5	Robert Wilson	+1234567899	+1234567899	123 Maple Street, Springfield, IL, 62701	No known allergies	1	2026-02-12 11:21:42	2026-02-12 11:21:42
<input type="checkbox"/>	2	CH2024001345	Oliver Wilson	2024-06-20	MALE	3.50	52.00	0+	5	Robert Wilson	+1234567899	+1234567899	123 Maple Street, Springfield, IL, 62701	No known allergies	1	2026-02-12 11:21:42	2026-02-12 11:21:42
<input type="checkbox"/>	3	CH2023001456	Sophia Brown	2023-01-10	FEMALE	3.00	49.00	A+	6	James Brown	+1234567810	+1234567810	456 Oak Avenue, Chicago, IL, 60614	Lactose intolerance	1	2026-02-12 11:21:42	2026-02-12 11:21:42
<input type="checkbox"/>	4	CH2024001567	Noah Brown	2024-08-05	MALE	3.80	53.00	A+	6	James Brown	+1234567810	+1234567810	456 Oak Avenue, Chicago, IL, 60614	No known allergies	1	2026-02-12 11:21:42	2026-02-12 11:21:42
<input type="checkbox"/>	5	CH2023001678	Isabella Davis	2022-11-02	FEMALE	3.30	51.00	B+	7	William Davis	+1234567811	+1234567811	789 Pine Road, Peoria, IL, 61602	Peanut allergy	1	2026-02-12 11:21:42	2026-02-12 11:21:42
<input type="checkbox"/>	6	CH2025001789	Liam Davis	2025-01-02	MALE	3.60	51.50	B+	7	William Davis	+1234567811	+1234567811	789 Pine Road, Peoria, IL, 61602	No known allergies	1	2026-02-12 11:21:42	2026-02-12 11:21:42
<input type="checkbox"/>	7	CH2023001890	Ava Miller	2023-07-08	FEMALE	3.10	49.50	AB+	8	Richard Miller	+1234567812	+1234567812	321 Elm Street, Rockford, IL, 61101	Eczema	1	2026-02-12 11:21:42	2026-02-12 11:21:42
<input type="checkbox"/>	8	CH2024001991	Ethan Moore	2024-02-14	MALE	3.40	51.00	O-	9	Charles Moore	+1234567813	+1234567813	654 Birch Lane, Naperville, IL, 60540	No known allergies	1	2026-02-12 11:21:42	2026-02-12 11:21:42
<input type="checkbox"/>	9	CH2023001992	Mucyo Abel Rwabugiri	2026-02-04	FEMALE	1.20	0.50	A-	10	Manzi	0788888884	0788888884	Kicukiro	None	1	2026-02-12 11:00:36	2026-02-12 15:59:30
<input type="checkbox"/>	10	CH2025001993	Mucyo ANGE	2026-02-06	FEMALE	1.20	0.50	B+	10	mUKIZA	0788888884	0788888884	Kicukiro	None	1	2026-02-12 16:05:29	2026-02-12 16:05:29
<input type="checkbox"/>	11	CH2023001994	Munyana ANGE	2026-02-06	FEMALE	1.20	0.50	B+									

### 3.2 Read Operation

The Read operation retrieves all records from the database using an SQL SELECT statement. ResultSet is used to iterate through the retrieved records and display them in the console.

### Select All Information of Children

**Mucyo Abel Rwabugiri** (CH2026508324)

**Basic Information**

- Full Name: Mucyo Abel Rwabugiri
- Date of Birth: 2026-02-04
- Age: 0 years, 0 months (0 months total)
- Gender: FEMALE
- Blood Group: A-

**Birth Information**

- Birth Weight: 1.2 kg
- Birth Height: 0.5 cm
- Profile ID: CH2026508324

**Parent Information**

- Mother: Akisa Divine
- Mother's Phone: 0792502568
- Father: Manzi
- Father's Phone: 0788888884

**Contact Information**

- Emergency Contact: 0788864532
- Address: Kicukiro

**Medical Information**

- None

[Edit Information](#) | [Delete Child](#)

### 3.3 Update Operation

The Update operation modifies existing records using an SQL UPDATE statement with a WHERE clause to target a specific record by ID.

Before Update(Mucyo Abel)

The screenshot shows two windows side-by-side. The left window is a MySQL Workbench interface displaying the 'children' table. The table has 9 rows of data, each representing a child with details like name, birth date, gender, and medical history. The right window is a CHMS (Child Health Management System) application dashboard for a user named Akisa Divine. The dashboard includes a welcome message, summary statistics (1 child, 0 upcoming appointments, 0 notifications), and sections for 'My Children' (listing Mucyo Abel) and 'Upcoming Appointments' (showing no upcoming appointments).

child_id	unique_profile_id	full_name	date_of_birth	gender	birth_weight	birth_height	blood_group	mother_id	father_name	father_phone	emergency_contact	address	medical_history	is_active	created_at	updated_at
1	CH0202001234	Emma Wilson	2023-03-15	FEMALE	3.20	50.00	O+	5	Robert Wilson	+1234567890	+1234567890	123 Maple Street, Springfield, IL 62701	No known allergies	1	2023-02-12 12:12:42	2028-02-12 11:21:42
2	CH0202001235	Oliver Wilson	2024-06-20	MALE	3.50	52.00	O+	5	Robert Wilson	+1234567890	+1234567890	123 Maple Street, Springfield, IL 62701	No known allergies	1	2023-02-12 12:12:42	2028-02-12 11:21:42
3	CH0202001236	Sophia Brown	2023-01-10	FEMALE	3.00	49.00	A+	6	James Brown	+1234567890	+1234567890	456 Oak Avenue, Chicago, IL 60614	Larvae intolerant	1	2023-02-12 12:12:42	2028-02-12 11:21:42
4	CH0202001237	Noah Brown	2024-08-05	MALE	3.00	53.00	A+	6	James Brown	+1234567890	+1234567890	456 Oak Avenue, Chicago, IL 60614	No known allergies	1	2023-02-12 12:12:42	2028-02-12 11:21:42
5	CH0202001238	Isabella Davis	2023-11-22	FEMALE	3.30	51.00	B+	7	William Davis	+1234567891	+1234567891	789 Pine Road, Peoria, IL 61602	Peanut allergy	1	2023-02-12 12:12:42	2028-02-12 11:21:42
6	CH0202001239	Liam Davis	2025-01-12	MALE	3.60	51.50	B+	7	William Davis	+1234567891	+1234567891	789 Pine Road, Peoria, IL 61602	No known allergies	1	2023-02-12 12:12:42	2028-02-12 11:21:42
7	CH0202001240	Ava Miller	2023-07-08	FEMALE	3.10	49.50	AB+	8	Richard Miller	+1234567892	+1234567892	321 Elm Street, Rockford, IL 61101	Eczema	1	2023-02-12 12:12:42	2028-02-12 11:21:42
8	CH0202001241	Elton Moore	2024-02-14	MALE	3.40	51.00	O-	9	Charles Moore	+1234567893	+1234567893	654 Birch Lane, Naperville, IL 60540	No known allergies	1	2023-02-12 12:12:42	2028-02-12 11:21:42
9	CH0202001242	Mucyo Abel	2026-02-04	FEMALE	1.20	6.50	A-	10	Makos	0788888884	0788888884	Kikukiro	None	1	2023-02-12 13:00:36	2028-02-12 13:00:36

**CHMS Dashboard (Akisa Divine):**

- Welcome back, Akisa Divine!
- Total Children: 1
- Upcoming Appointments: 0
- New Notifications: 0
- Recent Notifications: No new notifications
- My Children: Mucyo Abel (ID: CH0202001242, DOB: 2026-02-04, Blood Group: A-, Birth Weight: 1.2 kg)
- Upcoming Appointments: No upcoming appointments

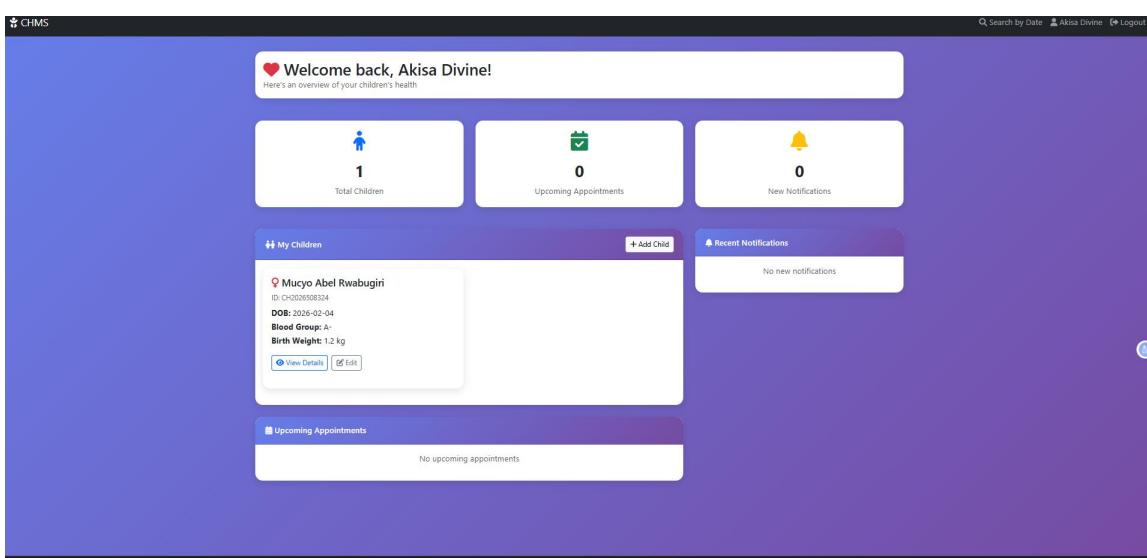
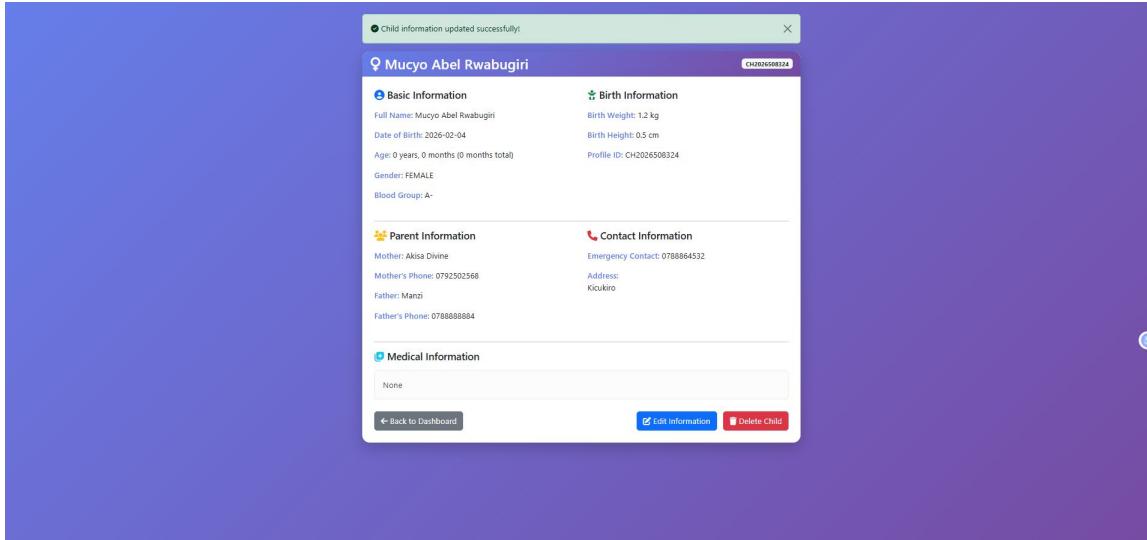
**Edit Child Information**

Child's Full Name *	Date of Birth *
Muyyo Abel	02/04/2026
Gender *	Blood Group
Female	A-
Birth Weight (kg) *	Birth Height (cm) *
1.2	0.5
Father's Name	Father's Phone
Manzi	0788888884
Emergency Contact *	
0788864532	
Address *	
Kicukiro	
Medical History / Allergies	
None	

After Update

**Edit Child Information**

Child's Full Name *	Date of Birth *
Muyyo Abel Rwanbugiri	02/04/2026
Gender *	Blood Group
Female	A-
Birth Weight (kg) *	Birth Height (cm) *
1.2	0.5
Father's Name	Father's Phone
Manzi	0788888884
Emergency Contact *	
0788864532	
Address *	
Kicukiro	
Medical History / Allergies	
None	



### 3.4 Delete Operation

The Delete operation removes records from the database using an SQL DELETE statement with a WHERE condition.

#### Table Creation

Table: children

Purpose: Store child profiles registered by mothers

```
CREATE TABLE children (
    child_id INT PRIMARY KEY AUTO_INCREMENT,
    unique_profile_id VARCHAR(50) UNIQUE NOT NULL,
```

```

full_name VARCHAR(100) NOT NULL,
date_of_birth DATE NOT NULL,
gender ENUM('MALE', 'FEMALE', 'OTHER') NOT NULL,
birth_weight DECIMAL(5,2) COMMENT 'Weight at birth in kg',
birth_height DECIMAL(5,2) COMMENT 'Height at birth in cm',
blood_group VARCHAR(5),
mother_id INT NOT NULL,
father_name VARCHAR(100),
father_phone VARCHAR(20),
emergency_contact VARCHAR(20),
address TEXT,
medical_history TEXT COMMENT 'Pre-existing conditions, allergies',
is_active BOOLEAN DEFAULT TRUE,
created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
FOREIGN KEY (mother_id) REFERENCES users(user_id) ON DELETE RESTRICT,
INDEX idx_mother_id (mother_id),
INDEX idx_dob (date_of_birth),
INDEX idx_active (is_active)
) ENGINE=InnoDB;

```

The child\_id field uniquely identifies each children record.

### Sample Data Insertion

Sample records were inserted for testing purposes:

```

INSERT INTO children (unique_profile_id, full_name, date_of_birth, gender, birth_weight,
birth_height, blood_group, mother_id, father_name, father_phone, emergency_contact, address,
medical_history) VALUES
('CH2023001234', 'Emma Wilson', '2023-03-15', 'FEMALE', 3.2, 50.0, '0+', 5, 'Robert Wilson',
'+1234567899', '+1234567899', '123 Maple Street, Springfield, IL 62701', 'No known allergies'),
('CH2024002345', 'Oliver Wilson', '2024-06-20', 'MALE', 3.5, 52.0, '0+', 5, 'Robert Wilson',
'+1234567899', '+1234567899', '123 Maple Street, Springfield, IL 62701', 'No known allergies'),

```

```

('CH2023003456', 'Sophia Brown', '2023-01-10', 'FEMALE', 3.0, 49.0, 'A+', 6, 'James Brown',
'+1234567810', '+1234567810', '456 Oak Avenue, Chicago, IL 60614', 'Lactose intolerant'),  

('CH2024004567', 'Noah Brown', '2024-08-05', 'MALE', 3.8, 53.0, 'A+', 6, 'James Brown',
'+1234567810', '+1234567810', '456 Oak Avenue, Chicago, IL 60614', 'No known allergies'),  

('CH2022005678', 'Isabella Davis', '2022-11-22', 'FEMALE', 3.3, 51.0, 'B+', 7, 'William Davis',
'+1234567811', '+1234567811', '789 Pine Road, Peoria, IL 61602', 'Peanut allergy'),  

('CH2025006789', 'Liam Davis', '2025-01-12', 'MALE', 3.6, 51.5, 'B+', 7, 'William Davis',
'+1234567811', '+1234567811', '789 Pine Road, Peoria, IL 61602', 'No known allergies'),  

('CH2023007890', 'Ava Miller', '2023-07-08', 'FEMALE', 3.1, 49.5, 'AB+', 8, 'Richard Miller',
'+1234567812', '+1234567812', '321 Elm Street, Rockford, IL 61101', 'Eczema'),  

('CH2024008901', 'Ethan Moore', '2024-02-14', 'MALE', 3.4, 51.0, 'O-', 9, 'Charles Moore',
'+1234567813', '+1234567813', '654 Birch Lane, Naperville, IL 60540', 'No known allergies');

```

The screenshot shows the phpMyAdmin interface for the 'chms\_db' database. The main area displays a table of 8 tables with their respective details:

Table	Action	Rows	Type	Collation	Size	Overhead
appointments	Browse  Structure  Search  Insert  Empty  Drop	18	InnoDB	utf8mb4_unicode_ci	88.0 Kib	-
audit_logs	Browse  Structure  Search  Insert  Empty  Drop	9	InnoDB	utf8mb4_unicode_ci	88.0 Kib	-
children	Browse  Structure  Search  Insert  Empty  Drop	10	InnoDB	utf8mb4_unicode_ci	88.0 Kib	-
growth_alerts	Browse  Structure  Search  Insert  Empty  Drop	2	InnoDB	utf8mb4_unicode_ci	96.0 Kib	-
health_records	Browse  Structure  Search  Insert  Empty  Drop	33	InnoDB	utf8mb4_unicode_ci	88.0 Kib	-
notifications	Browse  Structure  Search  Insert  Empty  Drop	15	InnoDB	utf8mb4_unicode_ci	96.0 Kib	-
users	Browse  Structure  Search  Insert  Empty  Drop	18	InnoDB	utf8mb4_unicode_ci	88.0 Kib	-
vaccinations	Browse  Structure  Search  Insert  Empty  Drop	28	InnoDB	utf8mb4_unicode_ci	64.0 Kib	-
<b>8 tables</b>	<b>Sum</b>	<b>117</b>	<b>InnoDB</b>	<b>utf8mb4_unicode_ci</b>	<b>656.0 Kib</b>	<b>0 B</b>

At the bottom left, there is a 'Create new table' form with fields for 'Table name' (set to '4') and 'Number of columns' (set to '4').

Database records before deletion(Mucyo Yannick)

Servers 127.0.0.1 Database chms\_db Table children

Browse Structure SQL Search Insert Export Import Privileges Operations Tracking Triggers

Showing rows 0 - 9 (10 total). Query took 0.0007 seconds.

SELECT \* FROM `children`

Profiling | Edit index | [Edit] | Explain SQL | Create PHP code | Refresh

Show all | Number of rows: 25 | Filter rows | Search this table | Sort by key: None

Extra options:

child_id	unique_profile_id	full_name	date_of_birth	gender	birth_weight	birth_height	blood_group	mother_id	father_name	father_phone	emergency_contact	address	medical_history	is_active	created_at	updated_at
1	CH2023001234	Emma Wilson	2023-01-15	FEMALE	3.20	50.00	O+	5	Robert Wilson	+1234567899	+1234567899	123 Maple Street, Springfield, IL 62701	No known allergies	1	2023-02-12 11:21:42	2023-02-12 11:21:42
2	CH2024002345	Oliver Wilson	2024-06-20	MALE	3.50	52.00	O+	5	Robert Wilson	+1234567899	+1234567899	123 Maple Street, Springfield, IL 62701	No known allergies	1	2023-02-12 11:21:42	2023-02-12 11:21:42
3	CH2023003456	Sophia Brown	2023-01-10	FEMALE	3.00	49.00	A+	8	James Brown	+1234567810	+1234567810	456 Oak Avenue, Chicago, IL 60615	Lactose intolerant	1	2023-02-12 11:21:42	2023-02-12 11:21:42
4	CH2023004567	Noah Brown	2023-06-05	MALE	3.80	53.00	A+	8	James Brown	+1234567810	+1234567810	456 Oak Avenue, Chicago, IL 60615	No known allergies	1	2023-02-12 11:21:42	2023-02-12 11:21:42
5	CH2023005678	Isabella Davis	2023-03-02	FEMALE	3.30	51.50	B+	7	William Davis	+1234567811	+1234567811	789 Pine Road, Peoria, IL 61602	Peanut allergy	1	2023-02-12 11:21:42	2023-02-12 11:21:42
6	CH2023006789	Liam Davis	2023-05-12	MALE	3.00	52.50	B+	7	William Davis	+1234567811	+1234567811	789 Pine Road, Peoria, IL 61602	No known allergies	1	2023-02-12 11:21:42	2023-02-12 11:21:42
7	CH2023007890	Ava Miller	2023-07-06	FEMALE	3.10	49.50	AB+	9	Richard Miller	+1234567812	+1234567812	521 Elm Street, Rockford, IL 61101	Eczema	1	2023-02-12 11:21:42	2023-02-12 11:21:42
8	CH2024008901	Ethan Moore	2024-03-14	MALE	3.40	51.00	O-	9	Charles Moore	+1234567813	+1234567813	654 Birch Lane, Naperville, IL 60540	No known allergies	1	2023-02-12 11:21:42	2023-02-12 11:21:42
9	CH2023009002	Macy Abel	2023-02-04	FEMALE	1.20	55.50	A-	10	Mandy Abel	0789888884	0789888884	Kukuleo	None	1	2023-02-12 13:00:36	2023-02-12 13:00:36
10	CH2023070044	Mucyo Yanick	2023-02-11	FEMALE	2.10	2.30	B+	10	Macy Abel	0789888884	0789888884	Kukuleo	None	1	2023-02-12 13:01:55	2023-02-12 13:01:55

Check all | With selected | Edit | Copy | Delete | Export

Show all | Number of rows: 25 | Filter rows | Search this table | Sort by key: None

Query results operations:

Print | Copy to clipboard | Export | Display chart | Create view

Bookmark this SQL query

Label:  Let every user access this bookmark

Bookmark this SQL query

Console

## Before Delete Children

CHMS

Welcome back, Akisa Divine!

Here's an overview of your children's health



2

Total Children



0

Upcoming Appointments



0

New Notifications

**My Children**

**Mucyo Yanick**  
ID: CH2023070044  
DOB: 2023-02-11  
Blood Group: B+  
Birth Weight: 2.1 kg

[View Details](#) | [Edit](#)

**Mucyo Abel**  
ID: CH2024009002  
DOB: 2024-03-14  
Blood Group: A-  
Birth Weight: 1.2 kg

[View Details](#) | [Edit](#)

**Recent Notifications**

No new notifications

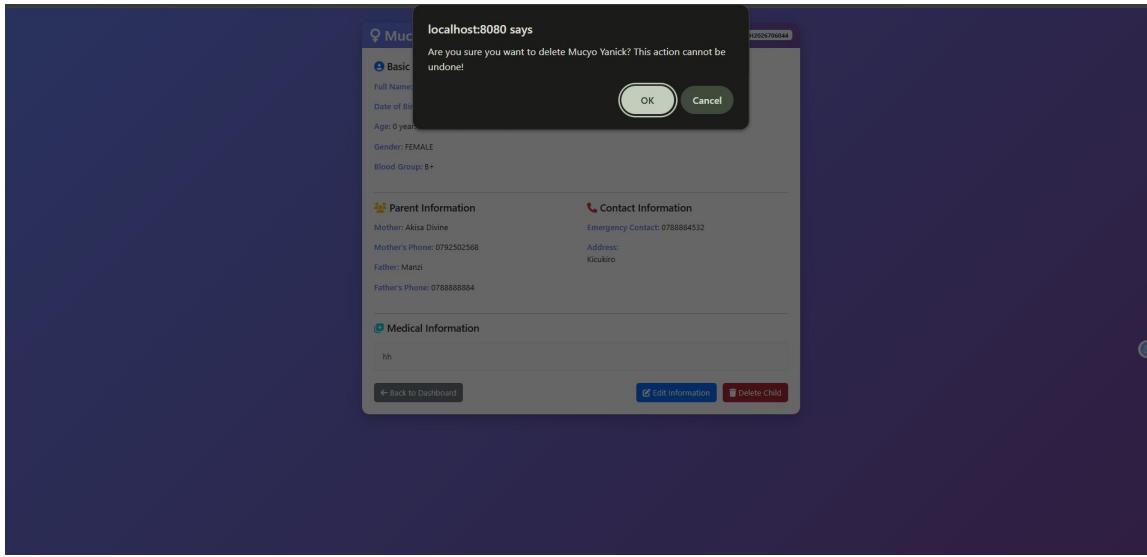
**Upcoming Appointments**

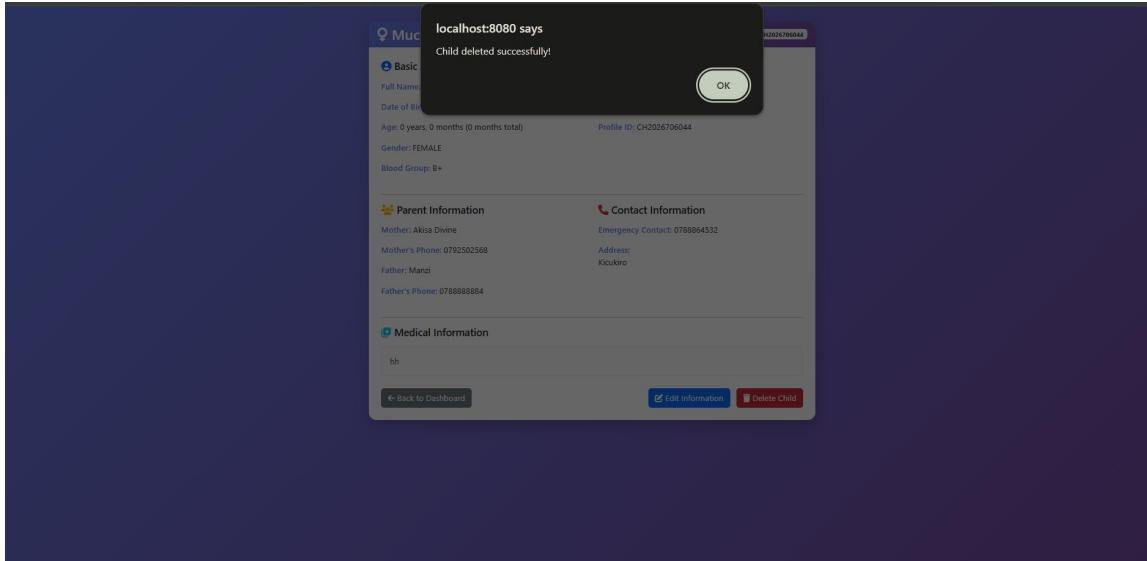
No upcoming appointments

 Mucyo Yanick CH2026706044

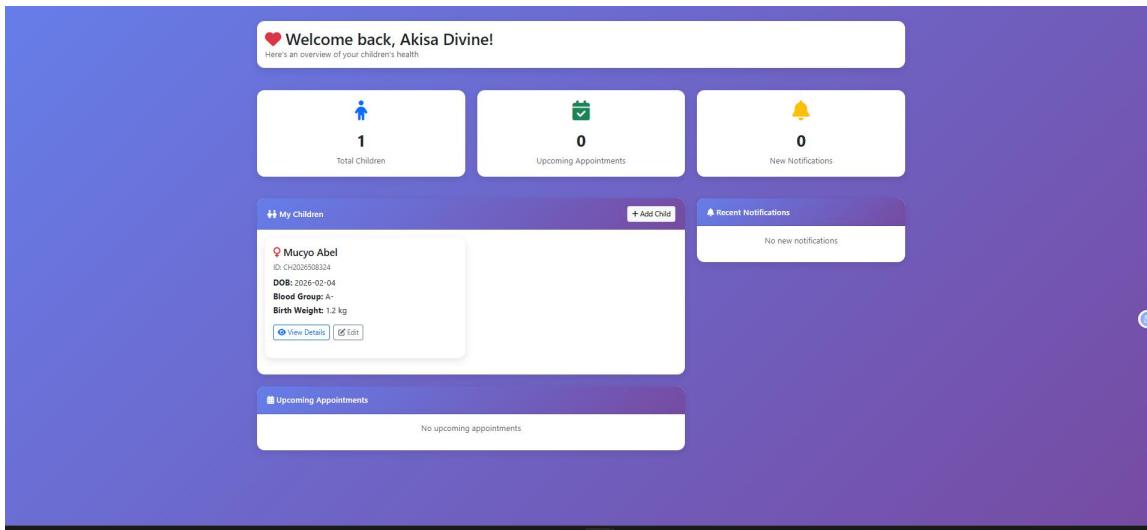
<b>Basic Information</b> Full Name: Mucyo Yanick Date of Birth: 2026-02-11 Age: 0 years, 0 months (0 months total) Gender: FEMALE Blood Group: B+	<b>Birth Information</b> Birth Weight: 2.1 kg Birth Height: 2.3 cm Profile ID: CH2026706044
<b>Parent Information</b> Mother: Akira Divine Mother's Phone: 0792502568 Father: Mandi Father's Phone: 0788888884	<b>Contact Information</b> Emergency Contact: 0788864532 Address: Kicukiro
<b>Medical Information</b> hh	

[← Back to Dashboard](#) [Edit information](#) [Delete Child](#)





After Delete Operation



Before executing the DELETE operation, the mother had 2 children but now remain with one children

#### 4. Search Feature (Select by Date)

The Search feature allows users to retrieve records based on a specific date. This functionality enhances the system by enabling filtered data retrieval instead of displaying all records.

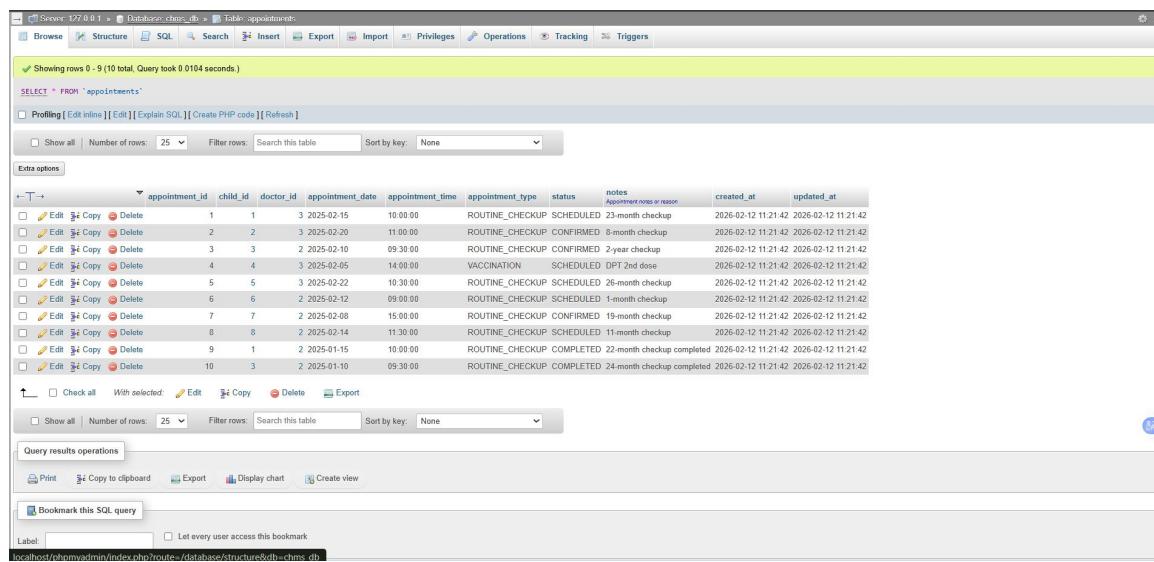
The search is implemented using an SQL SELECT query with a WHERE clause

## 5. Database Process

The date column used for searching is 'appointment\_date' of type DATE. The table contains multiple records with different dates to test the search functionality.

### Example Sample Records:

- 2025-02-15
- 2025-02-12
- 2025-02-22



A screenshot of the phpMyAdmin interface showing the 'appointments' table. The table has 10 rows of data. The columns are: appointment\_id, child\_id, doctor\_id, appointment\_date, appointment\_time, appointment\_type, status, notes, created\_at, and updated\_at. The data is as follows:

appointment_id	child_id	doctor_id	appointment_date	appointment_time	appointment_type	status	notes	created_at	updated_at
1	1	3	2025-02-15	10:00:00	ROUTINE_CHECKUP	SCHEDULED	23-month checkup	2026-02-12 11:21:42	2026-02-12 11:21:42
2	2	3	2025-02-20	11:00:00	ROUTINE_CHECKUP	CONFIRMED	8-month checkup	2026-02-12 11:21:42	2026-02-12 11:21:42
3	3	2	2025-02-10	09:30:00	ROUTINE_CHECKUP	CONFIRMED	2-year checkup	2026-02-12 11:21:42	2026-02-12 11:21:42
4	4	3	2025-02-05	14:00:00	VACCINATION	SCHEDULED	DPT 2nd dose	2026-02-12 11:21:42	2026-02-12 11:21:42
5	5	3	2025-02-22	10:30:00	ROUTINE_CHECKUP	SCHEDULED	26-month checkup	2026-02-12 11:21:42	2026-02-12 11:21:42
6	6	2	2025-02-12	09:00:00	ROUTINE_CHECKUP	SCHEDULED	1-month checkup	2026-02-12 11:21:42	2026-02-12 11:21:42
7	7	2	2025-02-08	15:00:00	ROUTINE_CHECKUP	CONFIRMED	19-month checkup	2026-02-12 11:21:42	2026-02-12 11:21:42
8	8	2	2025-02-14	11:30:00	ROUTINE_CHECKUP	SCHEDULED	11-month checkup	2026-02-12 11:21:42	2026-02-12 11:21:42
9	1	2	2025-01-15	10:00:00	ROUTINE_CHECKUP	COMPLETED	22-month checkup completed	2026-02-12 11:21:42	2026-02-12 11:21:42
10	3	2	2025-01-10	09:30:00	ROUTINE_CHECKUP	COMPLETED	24-month checkup completed	2026-02-12 11:21:42	2026-02-12 11:21:42

The date entered by the user is converted into java.sql.Date and passed as a parameter to the PreparedStatement. The ResultSet object retrieves matching records and displays them in the console.

### Java source code for Search feature

```

package com.cms.dao;

import com.cms.dao.AppointmentDAO;
import com.cms.dao.ChildDAO;
import com.cms.dao.UserDAO;
import com.cms.model.Appointment;
import com.cms.model.Child;
import com.cms.model.User;
import com.cms.util.SessionManager;
import com.google.appengine.api.datastore.Key;
import com.google.appengine.api.datastore.KeyFactory;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

import java.io.IOException;
import java.sql.Date;
import java.text.SimpleDateFormat;
import java.util.List;

/**
 * Servlet for searching records by date
 * Supports searching appointments, children (by DOB), and users (by registration date)
 */
@WebServlet("/search-by-date")
public class SearchByDateServlet extends HttpServlet {
    private static final Logger logger = LoggerFactory.getLogger(SearchByDateServlet.class);

    private final AppointmentDAO appointmentDAO = new AppointmentDAO();
    private final ChildDAO childDAO = new ChildDAO();
    private final UserDAO userDAO = new UserDAO();

    private final SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        // Check if user is logged in
        if (!SessionManager.isUserLoggedIn(request)) {
            response.sendRedirect(request.getContextPath() + "/login");
            return;
        }

        User loggedInUser = SessionManager.getloggedInUser(request);

        // Get search parameters
        String searchType = request.getParameter("type"); // appointments, children, users
        String dateStr = request.getParameter("date");
        String startDateStr = request.getParameter("startDate");
        String endDateStr = request.getParameter("endDate");

        try {
            public class SearchByDateServlet extends HttpServlet {
                protected void doGet(HttpServletRequest request, HttpServletResponse response) {
                    if (!SessionManager.isUserLoggedIn(request)) {
                        response.sendRedirect(request.getContextPath() + "/login");
                        return;
                    }

                    User loggedInUser = SessionManager.getloggedInUser(request);

                    // Get search parameters
                    String searchType = request.getParameter("type"); // appointments, children, users
                    String dateStr = request.getParameter("date");
                    String startDateStr = request.getParameter("startDate");
                    String endDateStr = request.getParameter("endDate");

                    try {
                        if (searchType == null || searchType.isEmpty()) {
                            Default view - show search form
                            request.setAttribute("searchType", "appointments");
                            request.getRequestDispatcher("/WEB-INF/views/search-by-date.jsp").forward(request, response);
                            return;
                        }

                        // Perform search based on type
                        switch (searchType.toLowerCase()) {
                            case "appointments":
                                searchAppointments(request, dateStr, startDateStr, endDateStr);
                                break;

                            case "children":
                                // Only admin can search all children
                                if (!loggedInUser.getRole().equals(User.Role.ADMIN) && !loggedInUser.getRole().equals(User.Role.DOCTOR)) {
                                    request.setAttribute("error", "Unauthorized access");
                                    break;
                                }
                                searchChildren(request, dateStr, startDateStr, endDateStr);
                                break;

                            case "users":
                                // Only admin can search users
                                if (!loggedInUser.getRole().equals(User.Role.ADMIN)) {
                                    request.setAttribute("error", "Unauthorized: Only admins can search users");
                                    break;
                                }
                                searchUsers(request, dateStr, startDateStr, endDateStr);
                                break;

                            default:
                                request.setAttribute("error", "Invalid search type");
                        }
                    } catch (Exception e) {
                        logger.error("Error during search: " + e.getMessage());
                        request.setAttribute("error", "Search failed: " + e.getMessage());
                    }
                }
            }
        
```

```

28  public class SearchByDateServlet extends HttpServlet {
29      protected void doGet(HttpServletRequest request, HttpServletResponse response)
30      {
31          request.setAttribute("searchType", "searchDate");
32          request.setAttribute("searchDate", dateStr);
33          request.setAttribute("startDate", startDateStr);
34          request.setAttribute("endDate", endDateStr);
35
36          catch (Exception e) {
37              logger.error("Error performing date search", e);
38              request.setAttribute("error", "Error: " + e.getMessage());
39          }
40
41          request.getRequestDispatcher("/WEB-INF/views/search-by-date.jsp").forward(request, response);
42      }
43
44
45      /**
46       * Search appointments by date or date range
47       */
48      private void searchAppointments(HttpServletRequest request, String dateStr,
49                                      String startDateStr, String endDateStr)
50      {
51          try {
52              List<Appointment> appointments;
53
54              if (dateStr != null && !dateStr.isEmpty())
55                  // Search by specific date
56                  searchByDate(dateStr);
57                  appointments = appointmentDAO.searchAppointmentsByDate(searchDate);
58                  request.setAttribute("searchMode", "single");
59                  logger.info("Searched appointments for date: {}, found: {}", dateStr, appointments.size());
60
61              else if (startDateStr != null && !startDateStr.isEmpty() &&
62                      endDateStr != null && !endDateStr.isEmpty())
63                  // Search by date range
64                  Date startDate = Date.valueOf(startDateStr);
65                  Date endDate = Date.valueOf(endDateStr);
66                  appointments = appointmentDAO.searchAppointmentsByDateRange(startDate, endDate);
67                  request.setAttribute("searchMode", "range");
68                  logger.info("Searched appointments from {} to {}, found: {}", startDateStr, endDateStr, appointments.size());
69
70              else {
71                  request.setAttribute("error", "Please provide either a specific date or a date range");
72                  return;
73              }
74
75              request.setAttribute("appointments", appointments);
76              request.setAttribute("resultCount", appointments.size());
77
78          } catch (IllegalArgumentException e) {
79              logger.error("Invalid date format", e);
80              request.setAttribute("error", "Invalid date format. Use YYYY-MM-DD");
81          }
82      }
83  }

```

The screenshot shows the CHMS application's dashboard. At the top, there is a welcome message: "Welcome back, Akisa Divine! Here's an overview of your children's health". Below this, there are three summary cards:

- Total Children:** 2
- Upcoming Appointments:** 0
- New Notifications:** 0

Below these cards, there are two sections:

- My Children:** Shows two children: Mucyo Yanick (ID: CH2026705044, DOB: 2026-02-11, Blood Group: B+, Birth Weight: 2.1 kg) and Mucyo Abel (ID: CH2026508324, DOB: 2026-02-04, Blood Group: A-, Birth Weight: 1.2 kg). Each child has "View Details" and "Edit" buttons.
- Recent Notifications:** Displays a message: "No new notifications".

## Input date used for searching

[← Back to Dashboard](#)

## 🔍 Search Records by Date

Search appointments, children, or users by date

Search Type:

Appointments by Date

Search by Specific Date:

mm/dd/yyyy

- OR -

Search by Date Range:

From: mm/dd/yyyy To: mm/dd/yyyy

**Search**

[← Back to Dashboard](#)

## 🔍 Search Records by Date

Search appointments, children, or users by date

Search Type:

Appointments by Date

Search by Specific Date:

02/12/2025

- OR -

Search by Date Range:

From: 02/01/2025 To: 02/28/2025

**Search**

## Output showing filtered results

The screenshot shows a search interface titled "Search Records by Date". The search type is set to "Appointments by Date". A specific date is selected as "02/12/2025". Below this, a date range is specified from "02/01/2025" to "02/28/2025". A single record is found, displayed in a table:

ID	Date	Time	Child	Doctor	Mother	Type	Status
6	2025-02-12	09:00:00	Liam Davis	Dr. Sarah Smith	Patricia Davis	ROUTINE_CHECKUP	SCHEDULED

## 7. Challenges Faced and Solutions

**Challenge 1:** Date format mismatch between user input and database.

Solution: Used `java.sql.Date.valueOf()` to ensure the correct DATE format.

**Challenge 2:** No results returned due to incorrect date values.

Solution: Verified stored database dates before performing the search.

**Challenge 3:** SQL syntax errors.

Tested the SQL query directly in the database before implementing it in Java.

## 8. Conclusion

The implementation of the Search (Select by Date) feature significantly enhances the CRUD system by enabling filtered data retrieval. The system now supports complete database management operations including Create, Read, Update, Delete, and Search functionalities. This project demonstrates effective use of JDBC, PreparedStatement, SQL queries, and proper database integration in Java backend development.