

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.

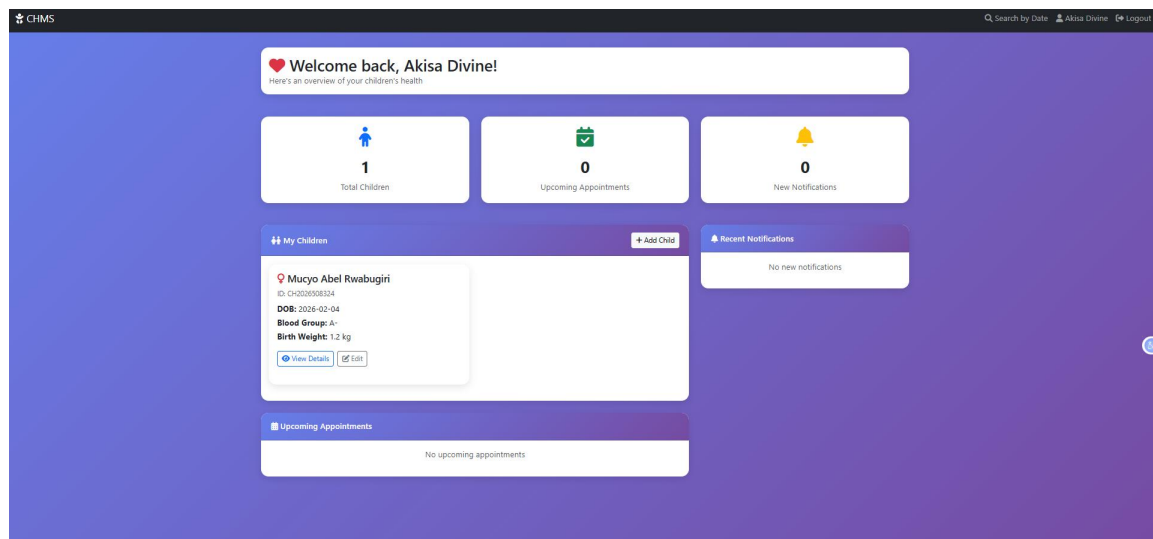
The appointment_date field was selected for the Search feature because it allows retrieval of transactions performed on a specific date, which is a common real-world requirement in financial systems.

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



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

SELECT * FROM "children"

Profiling

Edit view

SQL

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 (kg)(pounds)	birth_height (cm)(inches)	blood_group	mother_id	father_name	father_phone	emergency_contact	address	medical_history (Pneumonia, asthma, allergies)	is_active	created_at	updated_at			
<input type="checkbox"/>	Edit	Copy	Delete	1	CH2023001234	Emma Wilson	2024-03-15	FEMALE	3.20	50.00	O+	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"/>	Edit	Copy	Delete	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	2026-02-12 11:21:42	2026-02-12 11:21:42
<input type="checkbox"/>	Edit	Copy	Delete	3	CH2023003456	Sophia Brown	2023-01-10	FEMALE	3.00	48.00	A+	6	James Brown	+1234567810	+1234567810	456 Oak Avenue, Chicago, IL 60614	Lactose intolerant	1	2026-02-12 11:21:42	2026-02-12 11:21:42
<input type="checkbox"/>	Edit	Copy	Delete	4	CH2024004567	Noah Brown	2024-08-05	MALE	3.00	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"/>	Edit	Copy	Delete	5	CH2023005678	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	2026-02-12 11:21:42	2026-02-12 11:21:42
<input type="checkbox"/>	Edit	Copy	Delete	6	CH2025006789	Liam Davis	2025-01-12	MALE	3.60	51.00	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"/>	Edit	Copy	Delete	7	CH2023007890	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"/>	Edit	Copy	Delete	8	CH2024008901	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"/>	Edit	Copy	Delete	9	CH2026009012	Munyo Abel Ruzabughi	2026-02-04	FEMALE	1.20	0.50	A-	10	Mandi	0788888884	0788884532	Kioukro	None	1	2026-02-12 13:00:36	2026-02-12 15:59:30

☐ 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

After Insertion

Add New Child

Child's Full Name *

Munryana 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

mUNOZA

Father's Phone

0788888884

Emergency Contact *

0788864532

Address *

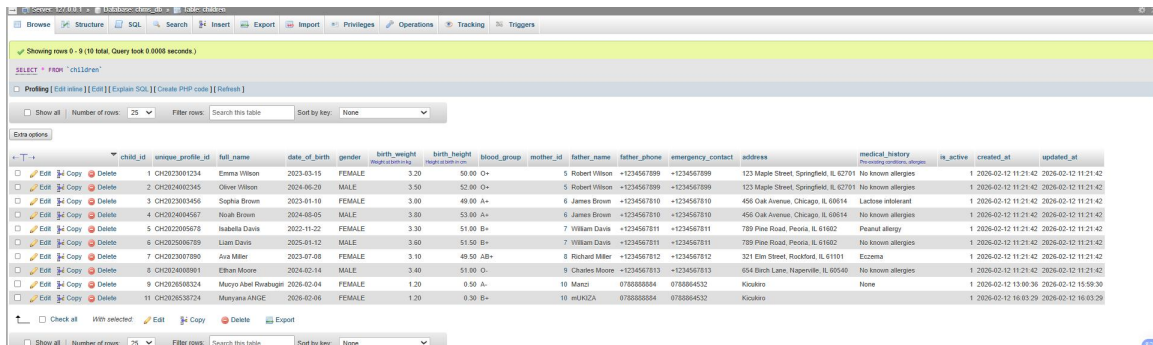
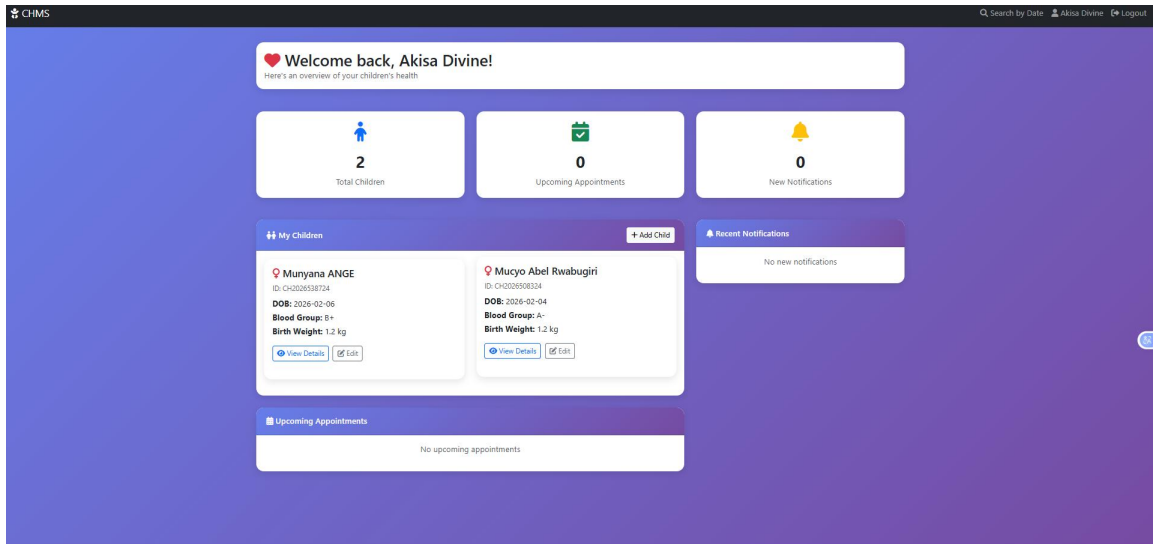
Kioukro

Medical History / Allergies

Eg, No known allergies, Asthma, Peanut allergy, etc.

Back to Dashboard

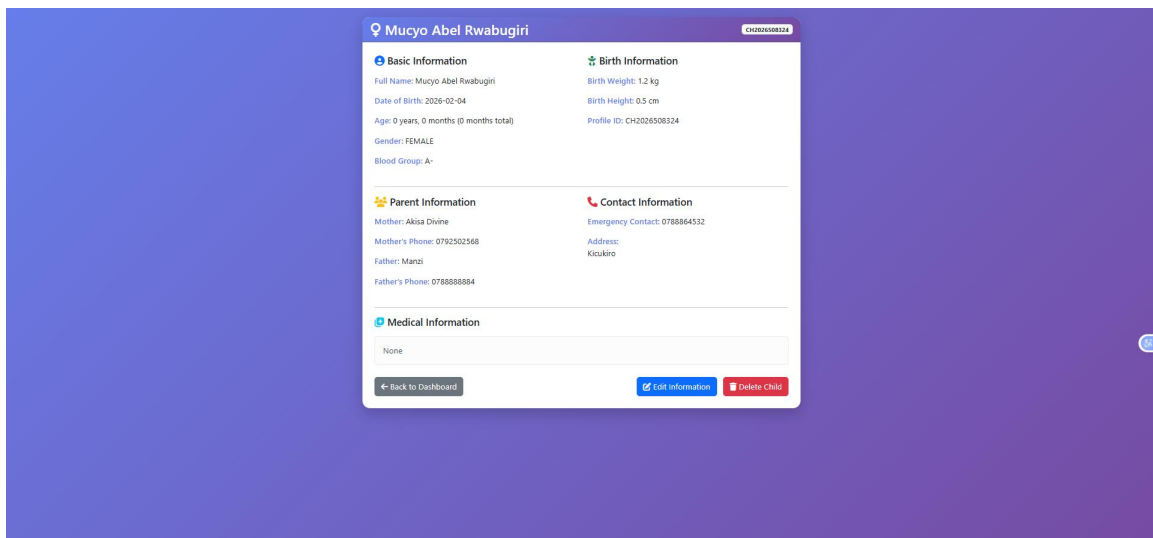
Add Child



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



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 displays a database management interface with a table of children records and a user dashboard.

Table: children

child_id	unique_profile_id	last_name	date_of_birth	gender	birth_weight (kg)	birth_height (cm)	blood_group	mother_id	father_name	father_phone	emergency_contact	address	medical_history	is_active	created_at	updated_at
1	CH020001234	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	2026-02-12 11:21:42	2026-02-12 11:21:42
2	CH020002345	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	2026-02-12 11:21:42	2026-02-12 11:21:42
3	CH020003456	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
4	CH020004567	Noah Brown	2024-05-05	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
5	CH020005678	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	2026-02-12 11:21:42	2026-02-12 11:21:42
6	CH020006789	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	2026-02-12 11:21:42	2026-02-12 11:21:42
7	CH020007890	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
8	CH020008901	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
9	CH020009012	Mucyo Abel	2026-02-04	FEMALE	1.20	6.50	A-	10	Mand	0788888884	0788884532	Kisumu	None	1	2026-02-12 13:00:36	2026-02-12 13:00:36

User Dashboard (CHMS)

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

- Total Children: 1
- Upcoming Appointments: 0
- New Notifications: 0

My Children (+ Add Child)

- Mucyo Abel**
ID: CH020009012
DOB: 2026-02-04
Blood Group: A-
Birth Weight: 1.2 kg
[View Details](#) [Edit](#)

Recent Notifications

No new notifications.

Upcoming Appointments

No upcoming appointments.

Edit Child Information

Child's Full Name *

Mucyo Abel

Date of Birth *

02/04/2026

Gender *

Female

Blood Group

A-

Birth Weight (kg) *

1.2

Birth Height (cm) *

0.5

Father's Name

Manzi

Father's Phone

0788888884

Emergency Contact *

0788884532

Address *

Kicukiro

Medical History / Allergies

None

Cancel

Save Changes

After Update

Edit Child Information

Child's Full Name *

Mucyo Abel Rwabugiri

Date of Birth *

02/04/2026

Gender *

Female

Blood Group

A-

Birth Weight (kg) *

1.2

Birth Height (cm) *

0.5

Father's Name

Manzi

Father's Phone

0788888884

Emergency Contact *

0788884532

Address *

Kicukiro

Medical History / Allergies

None

Cancel

Save Changes

Child information updated successfully!

Mucyo Abel Rwabugiri

CHILD026508324

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: Mami Father's Phone: 0788888884	Contact Information Emergency Contact: 0788884532 Address: Kicukiro
Medical Information None	

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

CHMS

Search by Date Akisa Divine Logout

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

<p>1 Total Children</p>	<p>0 Upcoming Appointments</p>	<p>0 New Notifications</p>
-----------------------------	------------------------------------	--------------------------------

My Children
[+ Add Child](#)

👤 Mucyo Abel Rwabugiri

ID: CH2026508324

DOB: 2026-02-04

Blood Group: A-

Birth Weight: 1.2 kg

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

Recent Notifications

No new notifications

Upcoming Appointments

No upcoming appointments

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, 'O+', 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, 'O+', 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'),

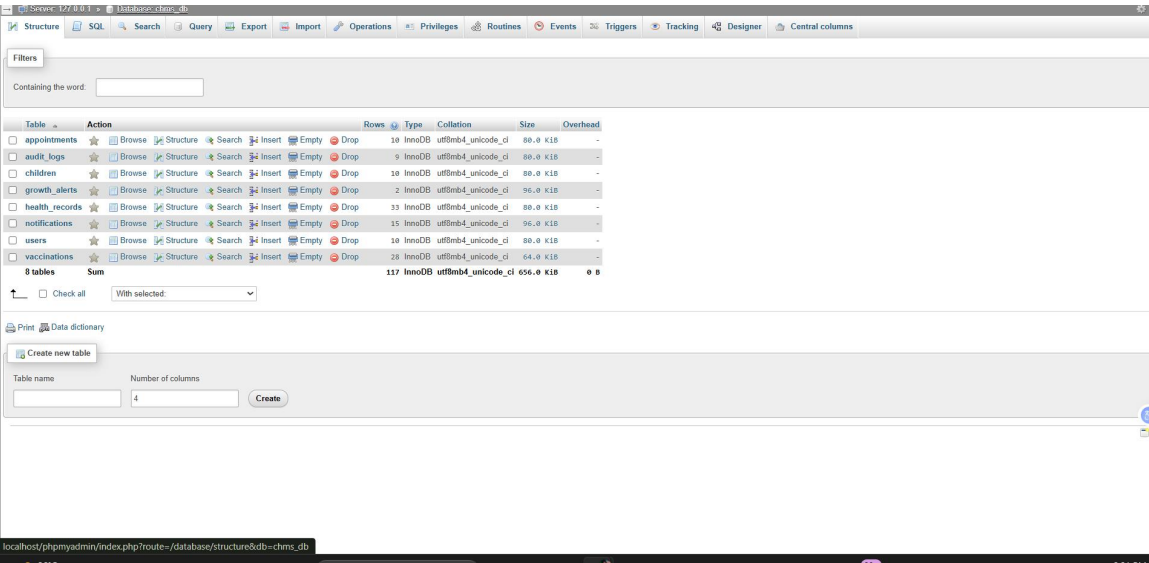
('CH2024004567', 'Noah Brown', '2024-08-05', 'MALE', 3.8, 53.0, 'A+', 6, 'James Brown',
'+1234567810', '+1234567810', '456 Oak Avenue, Chicago, IL 60614', 'No known allergies'),

('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');
```



Database records before deletion(Mucyo Yannick)

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

SELECT * FROM "children"

Pushing [Edit] [SQL] [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	unimsg_profile_id	first_name	date_of_birth	gender	birth_weight <small>(pounds/0.45)</small>	birth_height <small>(inches/2.54)</small>	blood_group	mother_id	father_name	father_phone	emergency_contact	address	medical_history <small>Phenotype/condition/usage</small>	is_active	created_at	updated_at
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	CH0203001234	Emma Wilson	2024-03-15	FEMALE	3.20	58.00	O+	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"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2	CH0204002345	Oliver Wilson	2024-06-29	MALE	3.50	62.00	O+	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"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	3	CH0203003456	Sophia Brown	2023-01-19	FEMALE	3.80	49.00	A+	6	James Brown	+1234567810	+1234567810	456 Oak Avenue, Chicago, IL 60614	Lactose intolerant	1	2026-02-12 11:21:42	2026-02-12 11:21:42
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	4	CH0204004567	Noah Brown	2024-05-05	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"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	5	CH0202005678	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	2026-02-12 11:21:42	2026-02-12 11:21:42
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	6	CH0205006789	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	2026-02-12 11:21:42	2026-02-12 11:21:42
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	7	CH0203007890	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"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	8	CH0204008901	Ethan Moore	2024-02-14	MALE	3.40	51.00	O-	9	Charles Moore	+1234567813	+1234567813	604 Birch Lane, Naperville, IL 60540	No known allergies	1	2026-02-12 11:21:42	2026-02-12 11:21:42
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	9	CH0206009012	Mucyo Abel	2026-02-04	FEMALE	1.20	6.50	A-	10	Mandi	0788888884	0788884032	Koukoro	None	1	2026-02-12 13:00:36	2026-02-12 13:00:36
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	10	CH0207009044	Mucyo Yanick	2026-02-11	FEMALE	2.10	2.30	B+	10	Mandi	0788888884	0788884032	Koukoro	hh	1	2026-02-12 13:01:55	2026-02-12 13:01:55

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

Before Delete Children

CHMS

Search by Date | Akisa Divine | Logout

Welcome back, Akisa Divine!

Here's an overview of your children's health

2

Total Children

0

Upcoming Appointments

0

New Notifications

My Children

+ Add Child

Mucyo Yanick

ID: CH0206709044

DOB: 2026-02-11

Blood Group: B+

Birth Weight: 2.1 kg

View Details | Edit

Mucyo Abel

ID: CH0206009034

DOB: 2026-02-04

Blood Group: A-

Birth Weight: 1.2 kg


View Details | Edit

Recent Notifications

No new notifications

Upcoming Appointments

No upcoming appointments



Mucyo Yanick

CH2026706044

Basic Information


Full Name: Mucyo Yanick

Date of Birth: 2026-02-11

Age: 0 years, 0 months (0 months total)

Gender: FEMALE

Blood Group: B+




Birth Information

Birth Weight: 2.1 kg

Birth Height: 2.3 cm

Profile ID: CH2026706044




Parent Information

Mother: Akisa Divine

Mother's Phone: 0792302368

Father: Manzi


Father's Phone: 0786888864



Contact Information


Emergency Contact: 0788864532

Address: Kicukino




Medical Information

hh




Lab Results


Lab Results



Back to Dashboard



Edit Information



Delete Child

localhost:8080 says

Are you sure you want to delete Mucyo Yanick? This action cannot be undone!

OK

Cancel

Basic

Full Name

Date of Birth

Age: 0 years

Gender: FEMALE

Blood Group: B+

Parent Information

Mother: Akisa Divine

Mother's Phone: 0792502568

Father: Manzi

Father's Phone: 0788888854

Contact Information

Emergency Contact: 0788884532

Address: Kikukiro

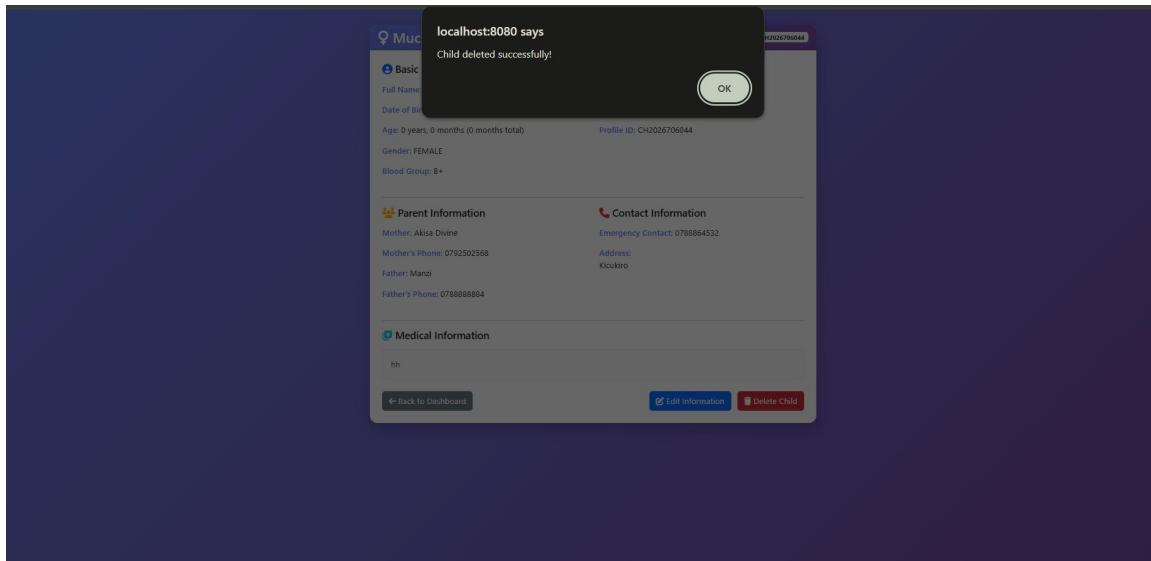
Medical Information

hh

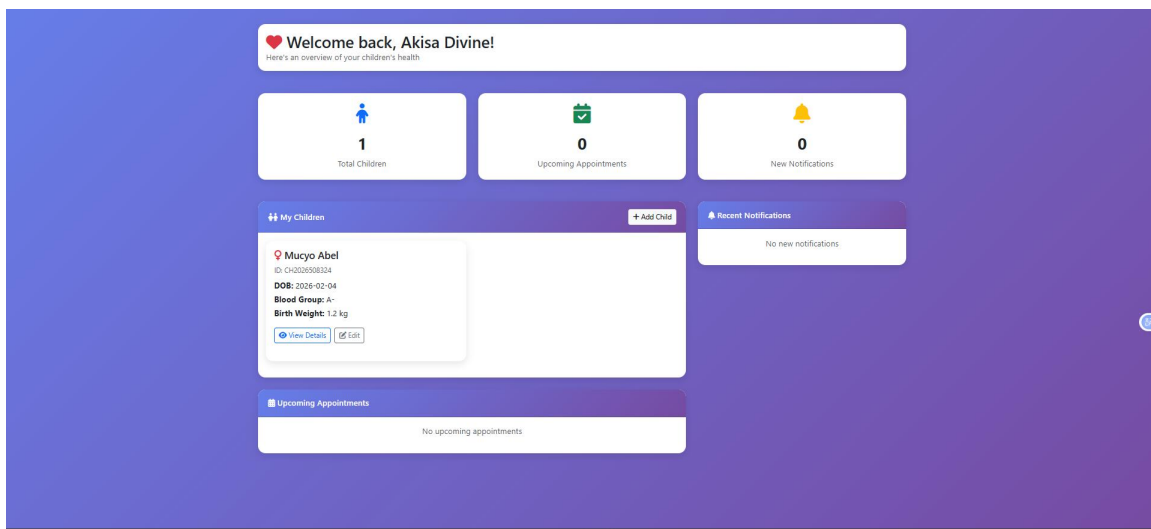
← Back to Dashboard

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.

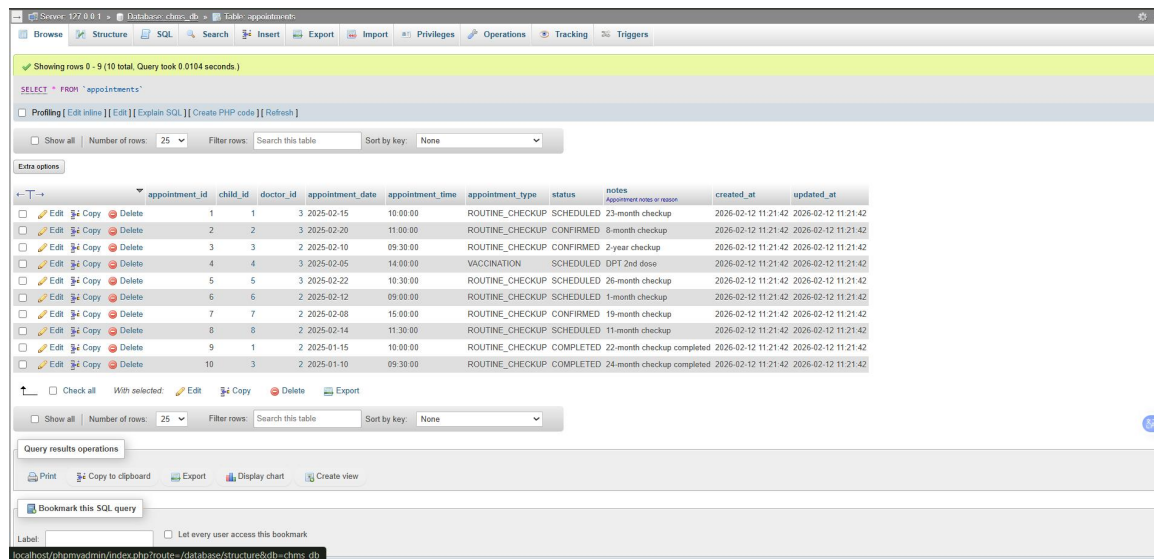
The search is implemented using an SQL SELECT query with a WHERE clause

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



The screenshot shows a database management interface with a table named 'appointments'. The table contains 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	5	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

```

1 package com.chms.servlet;
2
3 import com.chms.dao.AppointmentDAO;
4 import com.chms.dao.ChildDAO;
5 import com.chms.dao.UserDAO;
6 import com.chms.model.Appointment;
7 import com.chms.model.Child;
8 import com.chms.model.User;
9 import com.chms.util.SessionManager;
10 import jakarta.servlet.ServletException;
11 import jakarta.servlet.annotation.WebServlet;
12 import jakarta.servlet.http.HttpServlet;
13 import jakarta.servlet.http.HttpServletRequest;
14 import jakarta.servlet.http.HttpServletResponse;
15 import org.slf4j.Logger;
16 import org.slf4j.LoggerFactory;
17
18 import java.io.IOException;
19 import java.sql.Date;
20 import java.text.SimpleDateFormat;
21 import java.util.List;

```

```

22 /**
23  * Servlet for searching records by date
24  * Supports searching appointments, children (by DOB), and users (by registration date)
25  */
26 @WebServlet("/search-by-date")
27 public class SearchByDateServlet extends HttpServlet {
28     private static final Logger logger = LoggerFactory.getLogger(SearchByDateServlet.class);
29
30     private final AppointmentDAO appointmentDAO = new AppointmentDAO();
31     private final ChildDAO childDAO = new ChildDAO();
32     private final UserDAO userDAO = new UserDAO();
33
34     private final SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");
35
36     @Override
37     protected void doGet(HttpServletRequest request, HttpServletResponse response)
38         throws ServletException, IOException {
39
40         // Check if user is logged in
41         if (!SessionManager.isUserLoggedIn(request)) {
42             response.sendRedirect(request.getContextPath() + "/login");
43             return;
44         }
45
46         User loggedInUser = SessionManager.getLoggedInUser(request);
47
48         // Get search parameters
49         String searchType = request.getParameter("type"); // appointments, children, users
50         String dateStr = request.getParameter("date");
51         String startDateStr = request.getParameter("startDate");
52         String endDateStr = request.getParameter("endDate");
53
54         try {

```

```

55         public class SearchByDateServlet extends HttpServlet {
56             protected void doGet(HttpServletRequest request, HttpServletResponse response)
57                 throws ServletException, IOException {
58
59                 if (!SessionManager.isUserLoggedIn(request)) {
60                     response.sendRedirect(request.getContextPath() + "/login");
61                     return;
62                 }
63
64                 User loggedInUser = SessionManager.getLoggedInUser(request);
65
66                 // Get search parameters
67                 String searchType = request.getParameter("type"); // appointments, children, users
68                 String dateStr = request.getParameter("date");
69                 String startDateStr = request.getParameter("startDate");
70                 String endDateStr = request.getParameter("endDate");
71
72                 try {
73                     if (searchType == null || searchType.isEmpty()) {
74                         // Default view - show search form
75                         request.setAttribute("searchType", "appointments");
76                         request.getRequestDispatcher("/WEB-INF/views/search-by-date.jsp").forward(request, response);
77                         return;
78                     }
79
80                     // Perform search based on type
81                     switch (searchType.toLowerCase()) {
82                         case "appointments":
83                             searchAppointments(request, dateStr, startDateStr, endDateStr);
84                             break;
85                         case "children":
86                             // Only admin and doctors can search all children
87                             if (!loggedInUser.getRole().equals(User.Role.ADMIN) &&
88                                 !loggedInUser.getRole().equals(User.Role.DOCTOR)) {
89                                 request.setAttribute("error", "Unauthorized access");
90                                 break;
91                             }
92                             searchChildren(request, dateStr, startDateStr, endDateStr);
93                             break;
94                         case "users":
95                             // Only admin can search users
96                             if (!loggedInUser.getRole().equals(User.Role.ADMIN)) {
97                                 request.setAttribute("error", "Unauthorized: Only admins can search users");
98                                 break;
99                             }
100                             searchUsers(request, dateStr, startDateStr, endDateStr);
101                             break;
102                         default:
103                             request.setAttribute("error", "Invalid search type");
104                     }
105
106                     request.setAttribute("searchType", searchType);
107                     request.setAttribute("searchDate", dateStr);
108                 } catch (Exception e) {
109                     logger.error("Search error: ", e);
110                     request.setAttribute("error", "Search failed");
111                 }
112
113                 request.getRequestDispatcher("/WEB-INF/views/search-by-date.jsp").forward(request, response);
114             }
115
116             private void searchAppointments(HttpServletRequest request, String dateStr, String startDateStr, String endDateStr) {
117                 // Search logic for appointments
118             }
119
120             private void searchChildren(HttpServletRequest request, String dateStr, String startDateStr, String endDateStr) {
121                 // Search logic for children
122             }
123
124             private void searchUsers(HttpServletRequest request, String dateStr, String startDateStr, String endDateStr) {
125                 // Search logic for users
126             }
127         }

```

```

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

```

CHMS

Search by Date Akisa Divine Log

Welcome back, Akisa Divine!

Here's an overview of your children's health

2

Total Children

0

Upcoming Appointments

0

New Notifications

My Children

+ Add Child

Mucyo Yanick

ID: CH2026706044

DOB: 2026-02-11

Blood Group: B+

Birth Weight: 2.1 kg

View Details
 Edit

Mucyo Abel

ID: CH2026508324

DOB: 2026-02-04

Blood Group: A-

Birth Weight: 1.2 kg

View Details
 Edit

Recent Notifications

No new notifications

Upcoming Appointments

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

54

Output showing filtered results

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

Search Results 1 record(s) found

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.