```
-- ==========================================

-- STEP 4️⃣: Create Triggers

-- ==========================================

DELIMITER //


CREATE TRIGGER set_default_status

BEFORE INSERT ON Complaint

FOR EACH ROW

BEGIN

  SET NEW.status = 'Pending';

  SET NEW.submit_date = CURDATE();

END;

//


CREATE TRIGGER update_status_on_response

AFTER INSERT ON Response

FOR EACH ROW

BEGIN

  UPDATE Complaint

  SET status = 'Resolved'

  WHERE complaint_id = NEW.complaint_id;

END;

//


CREATE TRIGGER log_complaint_deletion

AFTER DELETE ON Complaint

FOR EACH ROW

BEGIN
```

```
  INSERT INTO Complaint_Log (complaint_id, deleted_on)
  VALUES (OLD.complaint_id, NOW());
END;
//

DELIMITER ;

-- ==========================================
-- STEP 5️⃣ : Create Stored Procedures
-- ==========================================
DELIMITER //

-- Add Complaint
CREATE PROCEDURE AddComplaint(IN p_user_id INT, IN p_description TEXT)
BEGIN
  INSERT INTO Complaint(user_id, description) VALUES(p_user_id, p_description);
  SELECT * FROM Complaint
  WHERE user_id = p_user_id
  ORDER BY submit_date DESC
  LIMIT 5;
END;
//

-- Add Response
CREATE PROCEDURE AddResponse(IN p_complaint_id INT, IN p_admin_id INT, IN
p_response_text TEXT)
BEGIN
  INSERT INTO Response(complaint_id, admin_id, response_date, response_text)
  VALUES(p_complaint_id, p_admin_id, CURDATE(), p_response_text);
  SELECT * FROM Complaint WHERE complaint_id = p_complaint_id;
END;
//

-- Delete Complaint
CREATE PROCEDURE DeleteComplaint(IN p_complaint_id INT)
BEGIN
  DELETE FROM Complaint WHERE complaint_id = p_complaint_id;
  SELECT * FROM Complaint_Log WHERE complaint_id = p_complaint_id;
END;
//

-- Get Pending Complaints
CREATE PROCEDURE GetPendingComplaints()
BEGIN
  SELECT c.complaint_id, u.name, c.description
  FROM Complaint c
  JOIN User u ON c.user_id = u.user_id
  WHERE c.status = 'Pending';
END;
//

-- Get Resolved Complaints
CREATE PROCEDURE GetResolvedComplaints()
BEGIN
  SELECT c.complaint_id, u.name, c.description
  FROM Complaint c
  JOIN User u ON c.user_id = u.user_id
  WHERE c.status = 'Resolved';
END;
//

-- Get Complaints with Responses
CREATE PROCEDURE GetComplaintsWithResponses()
BEGIN
  SELECT u.name, c.description, r.response_text, r.response_date, a.name AS admin_name
  FROM Complaint c
  JOIN Response r ON c.complaint_id = r.complaint_id
  JOIN User u ON c.user_id = u.user_id
  JOIN Admin a ON r.admin_id = a.admin_id;
END;
//

-- Search Complaints by Status
CREATE PROCEDURE SearchComplaintsByStatus(IN p_status VARCHAR(20))
BEGIN
  SELECT c.complaint_id, u.name, c.description, c.status, c.submit_date
  FROM Complaint c
  JOIN User u ON c.user_id = u.user_id
  WHERE c.status = p_status
  ORDER BY c.submit_date DESC;
END;
//
```

```
-- Get Complaints by User
CREATE PROCEDURE GetComplaintsByUser(IN p_user_id INT)
BEGIN
  SELECT complaint_id, description, type, status, submit_date
  FROM Complaint
  WHERE user_id = p_user_id
  ORDER BY submit_date DESC;
END;
//

-- Get Complaints by Date Range
CREATE PROCEDURE GetComplaintsByDate(IN start_date DATE, IN end_date DATE)
BEGIN
  SELECT c.complaint_id, u.name, c.description, c.status, c.submit_date
  FROM Complaint c
  JOIN User u ON c.user_id = u.user_id
  WHERE c.submit_date BETWEEN start_date AND end_date
  ORDER BY c.submit_date DESC;
END;
//

-- Count Complaints by Status
CREATE PROCEDURE CountComplaintsByStatus()
BEGIN
  SELECT status, COUNT(*) AS total
  FROM Complaint
  GROUP BY status;
END;
```

```
//


DELIMITER ;
```