Phase 5: Apex Programming (Developer)

1. UpdateNextEligibleDonation Trigger

- Location: Object Manager > Blood Donor > Triggers
- Code snippet:

```
trigger UpdateNextEligibleDonation on Blood_Donor__c (before insert, before update) {
    for (Blood_Donor__c donor : Trigger.new) {
        if (donor.Last_Donation_Date__c != null) {
            donor.Next_Eligible_Donation__c = donor.Last_Donation_Date__c.addDays(90);
        }
    }
}
```

2. BloodDonorTriggerTest Class

- Location: Setup > Apex Classes
- Code snippet:

```
@isTest
public class BloodDonorTriggerTest {
    @isTest static void testUpdateNextEligibleDonation() {
        Blood_Donor__c donor = new Blood_Donor__c(
            Donor_Name__c          = 'Test Donor',
            Email__c               = 'testdonor@example.com',
            Last_Donation_Date__c  = Date.today().addDays(-100),
            Phone_Number__c        = '9876543210',
            Blood_Type__c          = 'A+'
        );
        insert donor;

        Blood_Donor__c insertedDonor = [
            SELECT Next_Eligible_Donation__c
```

```
            FROM Blood_Donor__c WHERE Id = :donor.Id
        ];
        System.assertEquals(Date.today().addDays(-10),
insertedDonor.Next_Eligible_Donation__c);
    }
}
```

3. BloodTypeHelper Utility Class

- Location: Setup > Apex Classes
- Code snippet:

```
public class BloodTypeHelper {
    public static Boolean isCompatible(String donorType, String recipientType) {
        if (donorType == 'O-' || recipientType == 'AB+') {
            return true;
        }
        if (donorType == recipientType) {
            return true;
        }
        if (donorType == 'O+' && (recipientType == 'A+' || recipientType == 'B+' ||
recipientType == 'AB+')) {
            return true;
        }
        return false;
    }
}
```

4. BloodRequestProcessor Trigger

- Location: Object Manager > Blood Request > Triggers
- Code snippet:

```
trigger BloodRequestProcessor on Blood_Request__c (after insert) {
    for (Blood_Request__c req : Trigger.new) {
        List<Blood_Donor__c> compatibleDonors = [
            SELECT Id, Name, Blood_Type__c
            FROM Blood_Donor__c
            WHERE Blood_Type__c = :req.Blood_Type_Required__c
            LIMIT 10
        ];
        System.debug('Found ' + compatibleDonors.size() + ' donors for request ' + req.Id);
    }
}
```

5. DonorEligibilityBatch Batch Class

- Location: Setup > Apex Classes
- Code snippet:

```
public class DonorEligibilityBatch implements Database.Batchable<sObject> {
    public Database.QueryLocator start(Database.BatchableContext BC) {
        return Database.getQueryLocator(
            'SELECT Id, Last_Donation_Date__c, Next_Eligible_Donation__c FROM
Blood_Donor__c WHERE Last_Donation_Date__c != null'
        );
    }
    public void execute(Database.BatchableContext BC, List<Blood_Donor__c> scope) {
        List<Blood_Donor__c> toUpdate = new List<Blood_Donor__c>();
        for (Blood_Donor__c d : scope) {
            Date expected = d.Last_Donation_Date__c.addDays(90);
            if (d.Next_Eligible_Donation__c != expected) {
                d.Next_Eligible_Donation__c = expected;
```

```
            toUpdate.add(d);
        }
    }
    if (!toUpdate.isEmpty()) update toUpdate;
  }
  public void finish(Database.BatchableContext BC) {
    System.debug('Donor eligibility batch complete');
  }
}
```

6. DailyInventoryScheduler Scheduled Class

- Location: Setup > Apex Classes
- Code snippet:

```
public class DailyInventoryScheduler implements Schedulable {
  public void execute(SchedulableContext sc) {
    List<Blood_Inventory__c> low = [
      SELECT Id, Blood_Type__c, Units_Available__c
      FROM Blood_Inventory__c
      WHERE Units_Available__c <= 5
    ];
    if (!low.isEmpty()) {
      List<Task> tasks = new List<Task>();
      for (Blood_Inventory__c i : low) {
        tasks.add(new Task(
          Subject = 'Low Stock Alert: ' + i.Blood_Type__c,
          Status = 'Not Started',
          Priority = 'High',
          ActivityDate = Date.today(),
          Description = 'Only ' + i.Units_Available__c + ' units left'
        ));
      }
      insert tasks;
```

```
        }
        Database.executeBatch(new DonorEligibilityBatch(), 200);
    }
}
```

7. NotificationService Future Method

- Location: Setup > Apex Classes
- Code snippet:

```
public class NotificationService {
    @future(callout=true)
    public static void sendHospitalNotification(Set<Id> requestIds) {
        List<Blood_Request__c> reqs = [
            SELECT Hospital_Name__c, Blood_Type_Required__c, Units_Required__c
            FROM Blood_Request__c
            WHERE Id IN :requestIds
        ];
        for (Blood_Request__c r : reqs) {
            System.debug('Notify: ' + r.Hospital_Name__c + ' needs ' + r.Units_Required__c + ' '
+ r.Blood_Type_Required__c);
        }
    }
    public static void createFollowUpTasks(List<Blood_Request__c> reqs) {
        List<Task> tasks = new List<Task>();
        for (Blood_Request__c r : reqs) {
            tasks.add(new Task(
                Subject = 'Follow up: ' + r.Hospital_Name__c,
                Status = 'Not Started',
                Priority = 'Normal',
                ActivityDate = Date.today().addDays(1),
                WhatId = r.Id
            ));
        }
```

```
        if (!tasks.isEmpty()) insert tasks;
    }
}
```

8. ApexTestSuite Comprehensive Test Class

- Location: Setup > Apex Classes
- Code snippet:

```
@isTest
public class ApexTestSuite {
    @TestSetup
    static void setupTestData() {
        List<Blood_Donor__c> ds = new List<Blood_Donor__c>();
        for (Integer i=0; i<5; i++) {
            ds.add(new Blood_Donor__c(
                Donor_Name__c          = 'Donor ' + i,
                Email__c               = 'donor'+i+'@test.com',
                Last_Donation_Date__c  = Date.today().addDays(-100),
                Phone_Number__c        = '9876543210',
                Blood_Type__c          = 'O+'
            ));
        }
        insert ds;
        insert new Blood_Inventory__c(Blood_Type__c='O+', Units_Available__c=3);
        insert new Blood_Inventory__c(Blood_Type__c='A+', Units_Available__c=15);
    }

    @isTest static void testBloodTypeHelper() {
        System.assert(BloodTypeHelper.isCompatible('O-','A+'));
        System.assert(!BloodTypeHelper.isCompatible('A+','B+'));
    }

    @isTest static void testBloodRequestTrigger() {
```

```
        Blood_Request__c r = new Blood_Request__c(
            Hospital_Name__c      = 'Test Hospital',
            Blood_Type_Required__c= 'O+',
            Units_Required__c     = 5,
            Request_Date__c       = Date.today(),
            Required_By_Date__c   = Date.today().addDays(7),
            Contact_Person__c     = 'Dr. Smith',
            Contact_Phone__c      = '9123456789'
        );
        insert r;
        System.assertNotEquals(null, r.Id);
    }

    @isTest static void testDonorEligibilityBatch() {
        Test.startTest();
        Database.executeBatch(new DonorEligibilityBatch(), 200);
        Test.stopTest();
        List<Blood_Donor__c> updated = [SELECT Next_Eligible_Donation__c FROM
Blood_Donor__c];
        System.assert(!updated.isEmpty());
    }

    @isTest static void testDailyInventoryScheduler() {
        Test.startTest();
        new DailyInventoryScheduler().execute(null);
        Test.stopTest();
        List<Task> alerts = [SELECT Id FROM Task WHERE Subject LIKE 'Low Stock
Alert%'];
        System.assert(!alerts.isEmpty());
    }

    @isTest static void testBloodInventoryTrigger() {
        Blood_Inventory__c inv = [SELECT Id, Units_Available__c FROM
Blood_Inventory__c WHERE Blood_Type__c='A+' LIMIT 1];
```

```
        inv.Units_Available__c = 4;
        Test.startTest(); update inv; Test.stopTest();
        System.assertEquals(4, inv.Units_Available__c);
    }
}
```

9. Test Execution

- Run all test classes (ApexTestSuite, BloodDonorTriggerTest, etc.).
- Confirm 100% pass rate.
- Ensure overall code coverage > 75%.