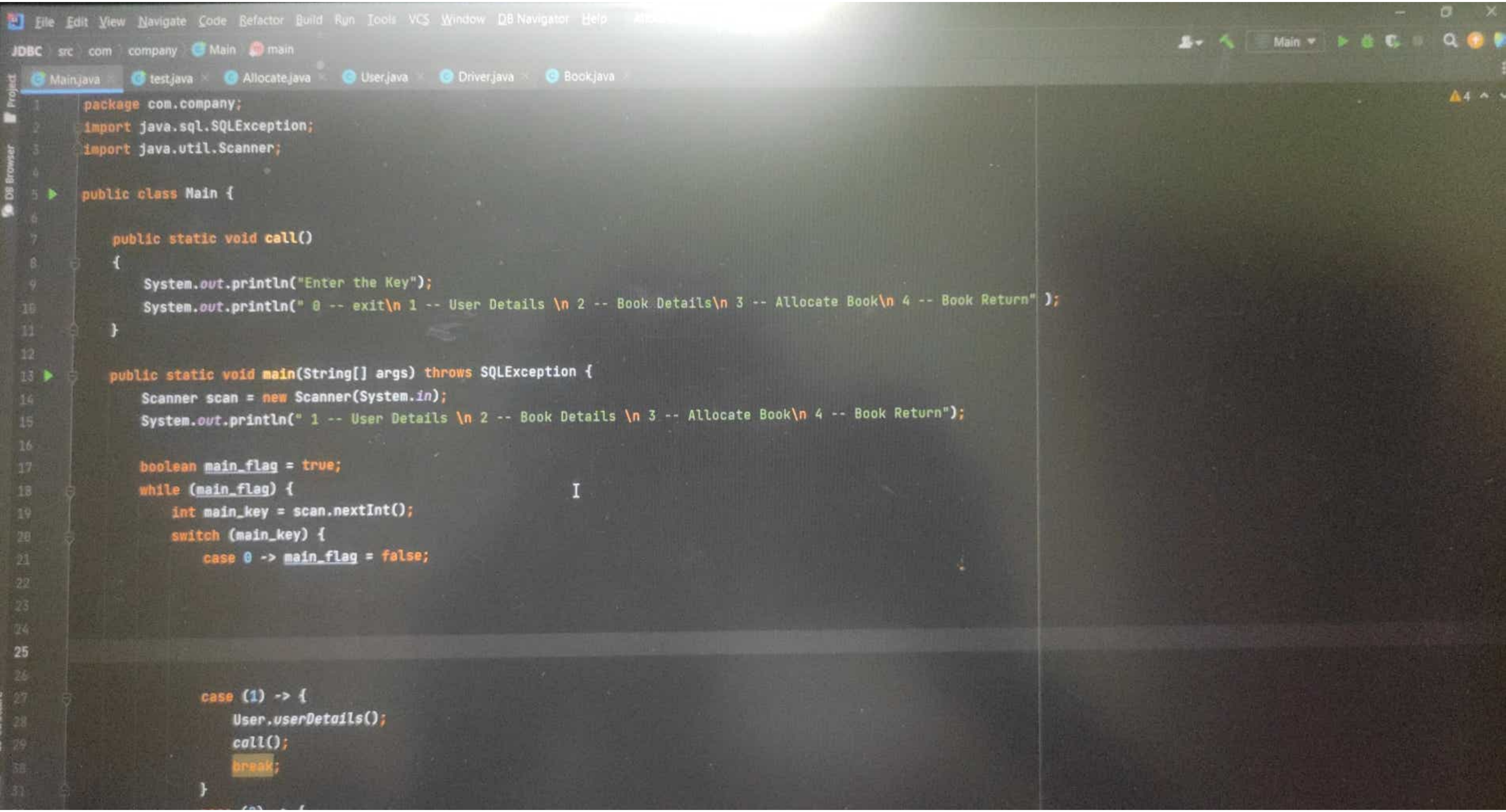
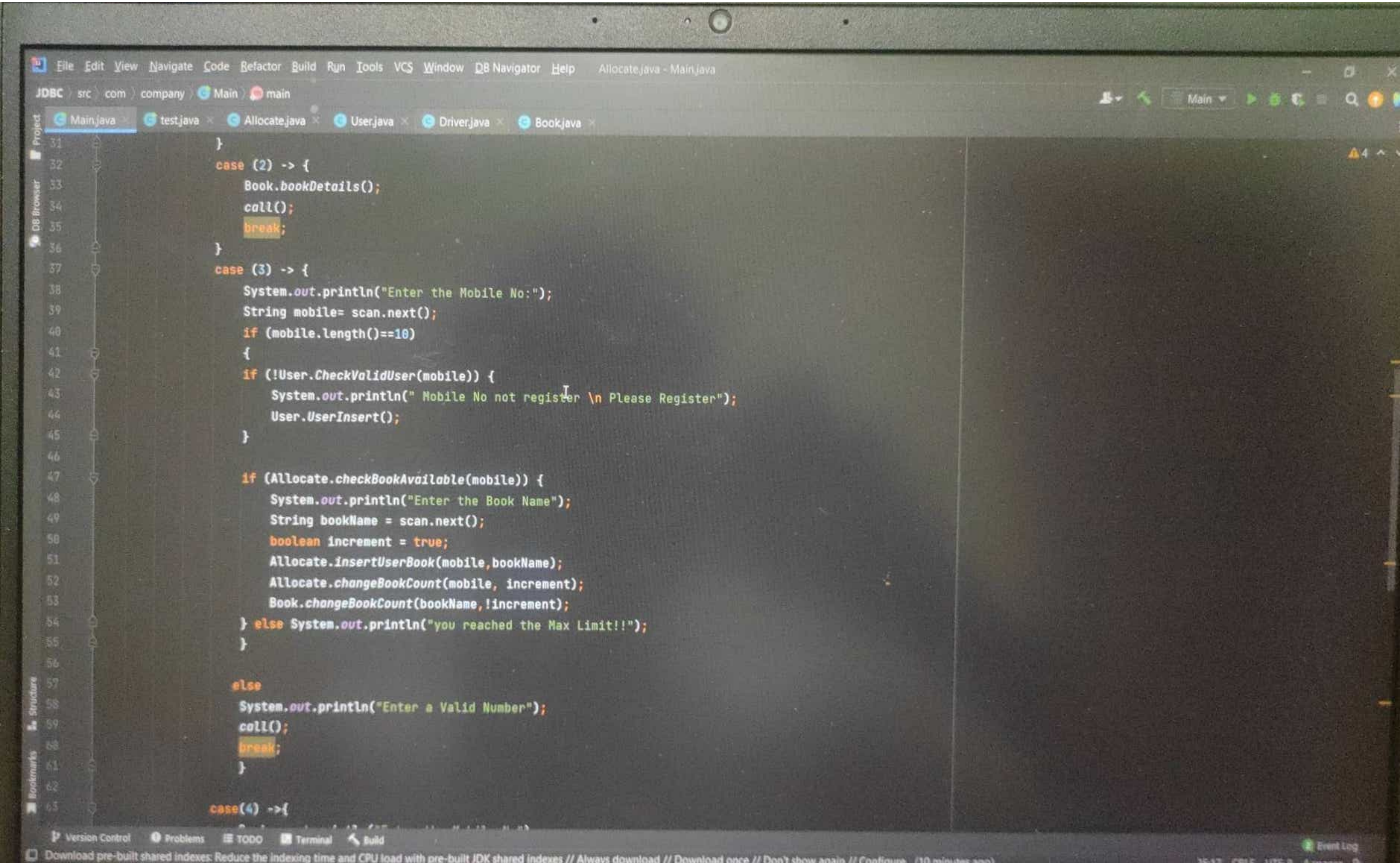
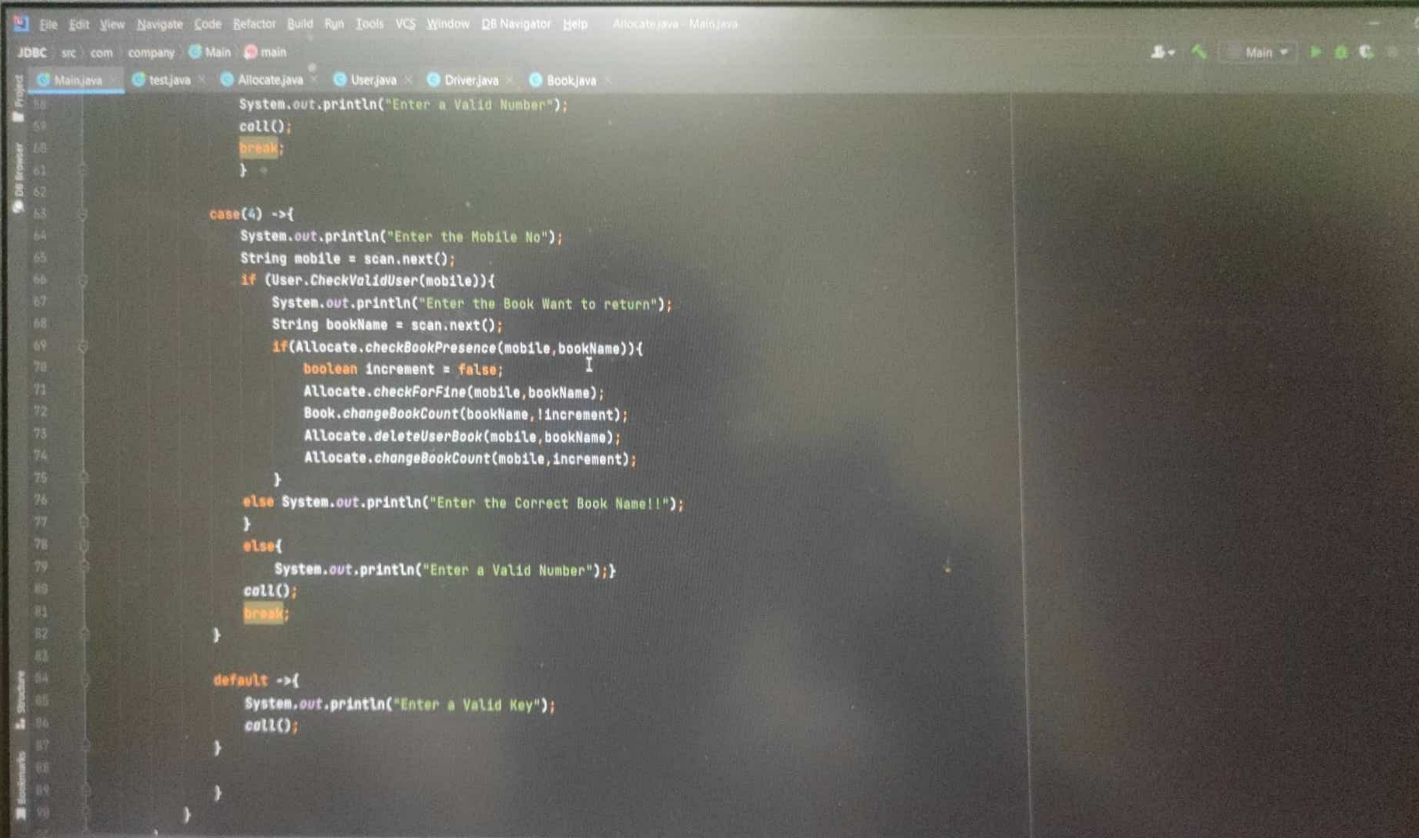

File Edit View Navigate Code Refactor Build Run Tools VCS Window DB Navigator Help
JDBC src com company Main main
Main.java
test.java
Allocate.java
User.java
Driver.java
Book.java
Project
DB Browser
Main
4

```
package com.company;
import java.sql.SQLException;
import java.util.Scanner;

public class Main {

    public static void call()
    {
        System.out.println("Enter the Key");
        System.out.println(" 0 -- exit\n 1 -- User Details \n 2 -- Book Details\n 3 -- Allocate Book\n 4 -- Book Return" );
    }

    public static void main(String[] args) throws SQLException {
        Scanner scan = new Scanner(System.in);
        System.out.println(" 1 -- User Details \n 2 -- Book Details \n 3 -- Allocate Book\n 4 -- Book Return");

        boolean main_flag = true;
        while (main_flag) {
            int main_key = scan.nextInt();
            switch (main_key) {
                case 0 -> main_flag = false;




                case (1) -> {
                    User.userDetails();
                    call();
                    break;
                }
```

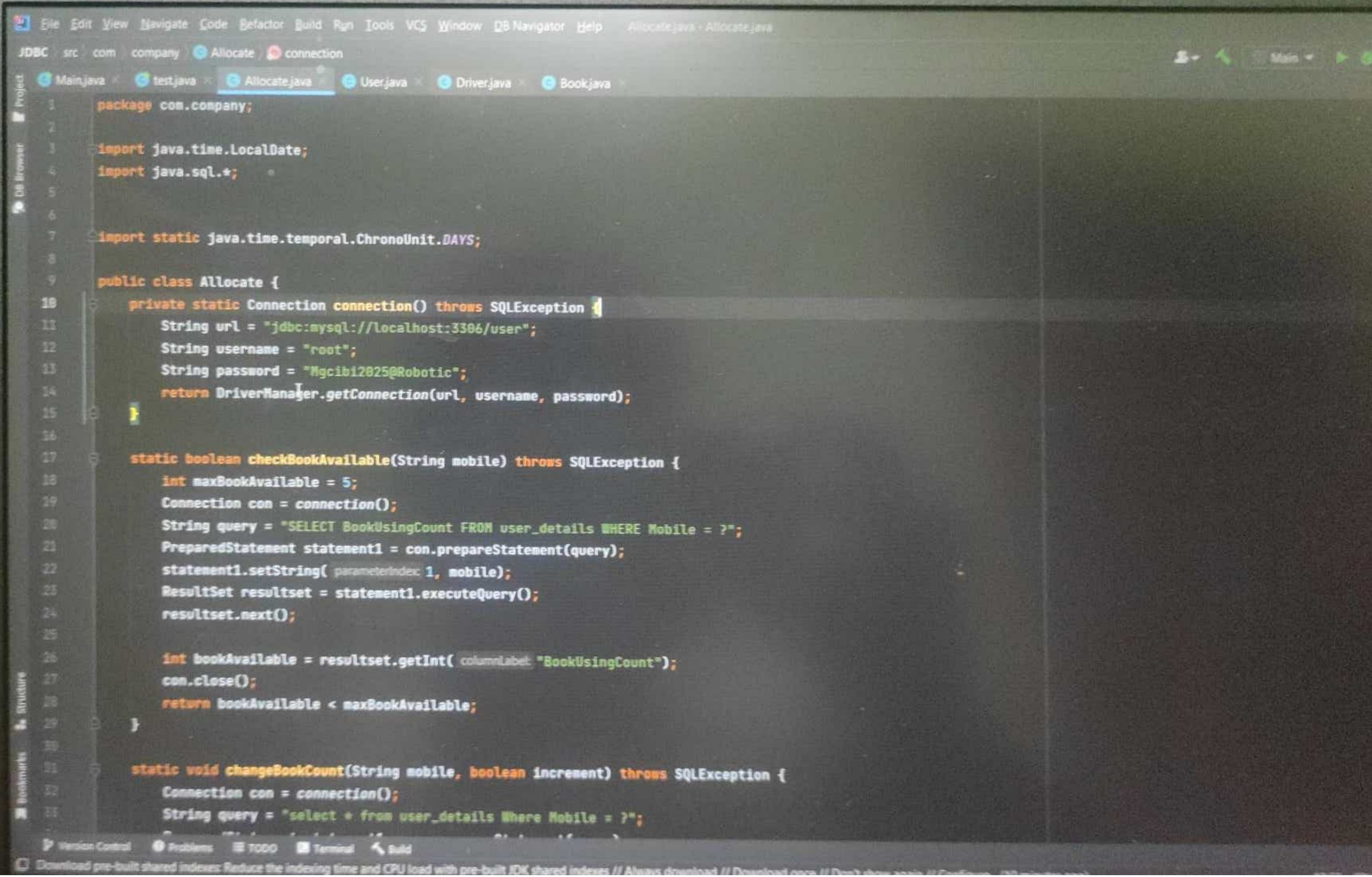


```java
            }
            case (2) -> {
                Book.bookDetails();
                call();
                break;
            }
            case (3) -> {
                System.out.println("Enter the Mobile No:");
                String mobile= scan.next();
                if (mobile.length()==10)
                {
                if (!User.CheckValidUser(mobile)) {
                    System.out.println(" Mobile No not register \n Please Register");
                    User.UserInsert();
                }

                if (Allocate.checkBookAvailable(mobile)) {
                    System.out.println("Enter the Book Name");
                    String bookName = scan.next();
                    boolean increment = true;
                    Allocate.insertUserBook(mobile,bookName);
                    Allocate.changeBookCount(mobile, increment);
                    Book.changeBookCount(bookName,!increment);
                } else System.out.println("you reached the Max Limit!!");
                }

              else
              System.out.println("Enter a Valid Number");
              call();
              break;
              }

           case(4) ->{
```

Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built JDK shared indexes // Always download // Download once // Don't show again // Configure... (10 minutes ago)

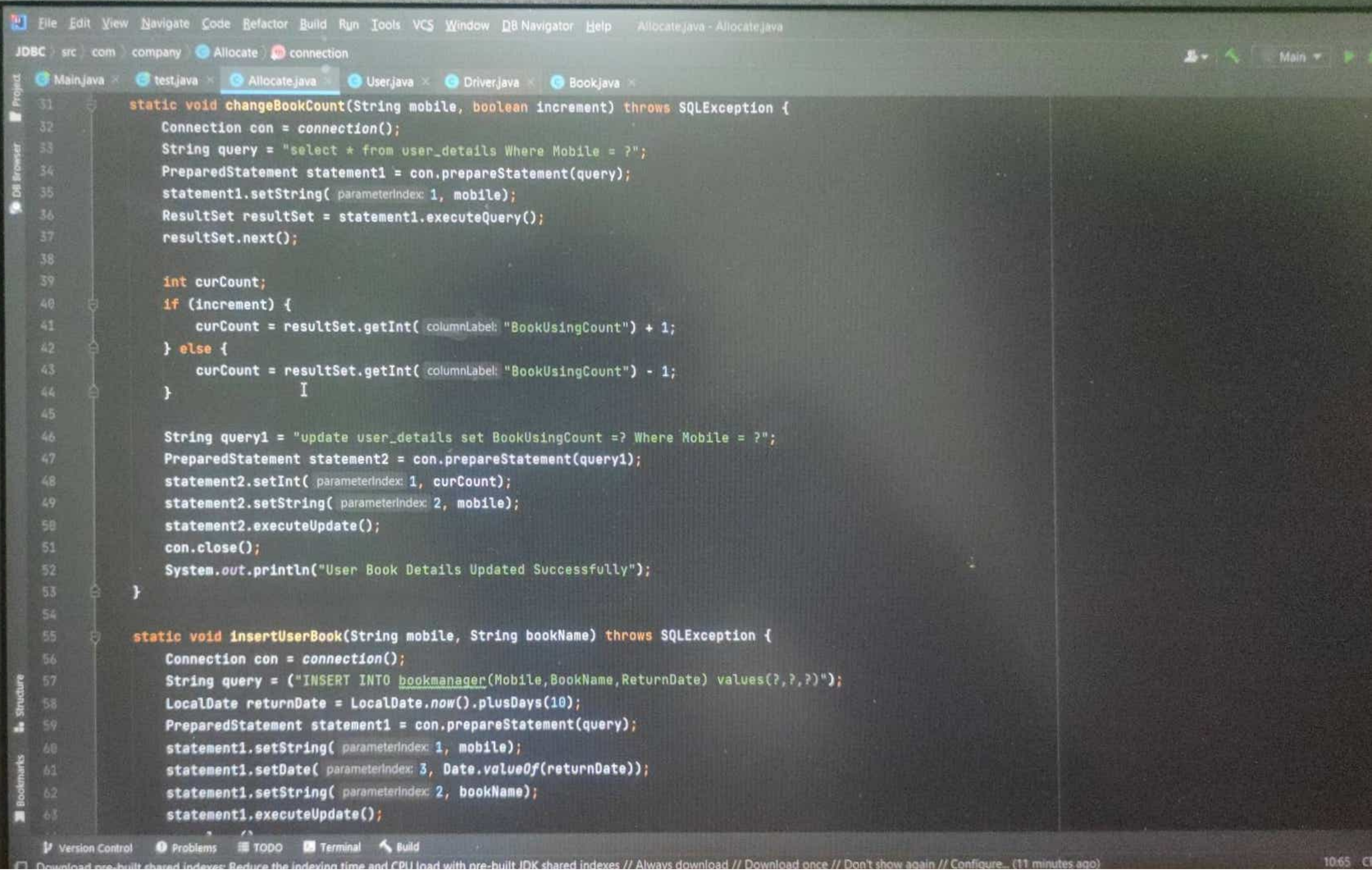


```
            System.out.println("Enter a Valid Number");
            call();
            break;
            }

        case(4) ->{
            System.out.println("Enter the Mobile No");
            String mobile = scan.next();
            if (User.CheckValidUser(mobile)){
                System.out.println("Enter the Book Want to return");
                String bookName = scan.next();
                if(Allocate.checkBookPresence(mobile,bookName)){
                    boolean increment = false;
                    Allocate.checkForFine(mobile,bookName);
                    Book.changeBookCount(bookName,!increment);
                    Allocate.deleteUserBook(mobile,bookName);
                    Allocate.changeBookCount(mobile,increment);
                }
            else System.out.println("Enter the Correct Book Name!!");
            }
            else{
                System.out.println("Enter a Valid Number");}
            call();
            break;
        }

        default ->{
            System.out.println("Enter a Valid Key");
            call();
        }

        }
    }
```


File Edit View Navigate Code Refactor Build Run Tools VCS Window DB Navigator Help
Allocate.java - Allocate.java
JDBC › src › com › company › Allocate › connection
Main.java
test.java
Allocate.java
User.java
Driver.java
Book.java
Project
DB Browser
Structure
Bookmarks
Main
Version Control
Problems
TODO
Terminal
Build

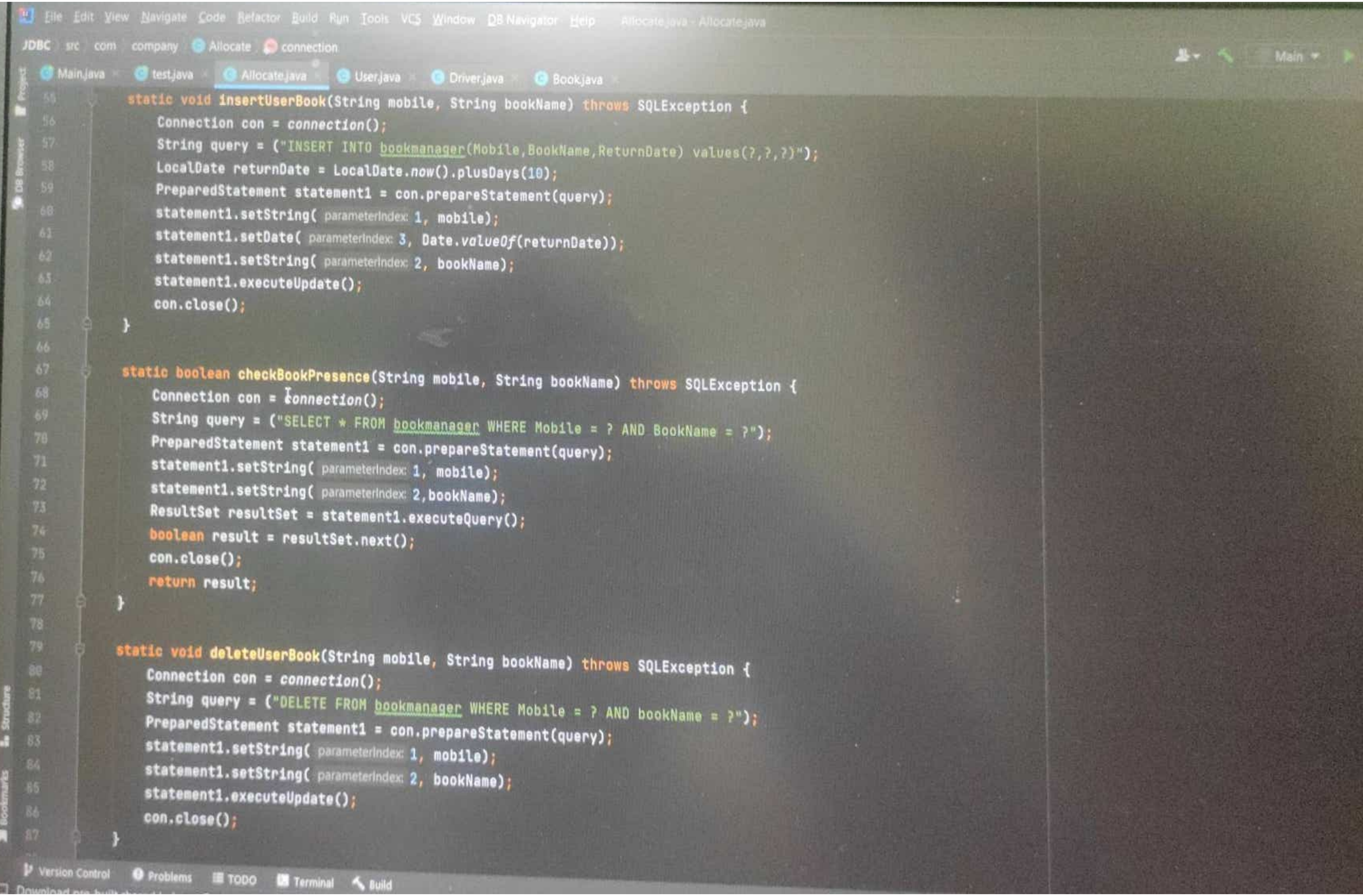
```java
package com.company;

import java.time.LocalDate;
import java.sql.*;


import static java.time.temporal.ChronoUnit.DAYS;

public class Allocate {
    private static Connection connection() throws SQLException {
        String url = "jdbc:mysql://localhost:3306/user";
        String username = "root";
        String password = "Mgcibi2025@Robotic";
        return DriverManager.getConnection(url, username, password);
    }

    static boolean checkBookAvailable(String mobile) throws SQLException {
        int maxBookAvailable = 5;
        Connection con = connection();
        String query = "SELECT BookUsingCount FROM user_details WHERE Mobile = ?";
        PreparedStatement statement1 = con.prepareStatement(query);
        statement1.setString( parameterIndex: 1, mobile);
        ResultSet resultset = statement1.executeQuery();
        resultset.next();

        int bookAvailable = resultset.getInt( columnLabel: "BookUsingCount");
        con.close();
        return bookAvailable < maxBookAvailable;
    }

    static void changeBookCount(String mobile, boolean increment) throws SQLException {
        Connection con = connection();
        String query = "select * from user_details Where Mobile = ?";
```

Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built JDK shared indexes // Always download // Download once // Don't show again // Configure... (30 minutes ago)

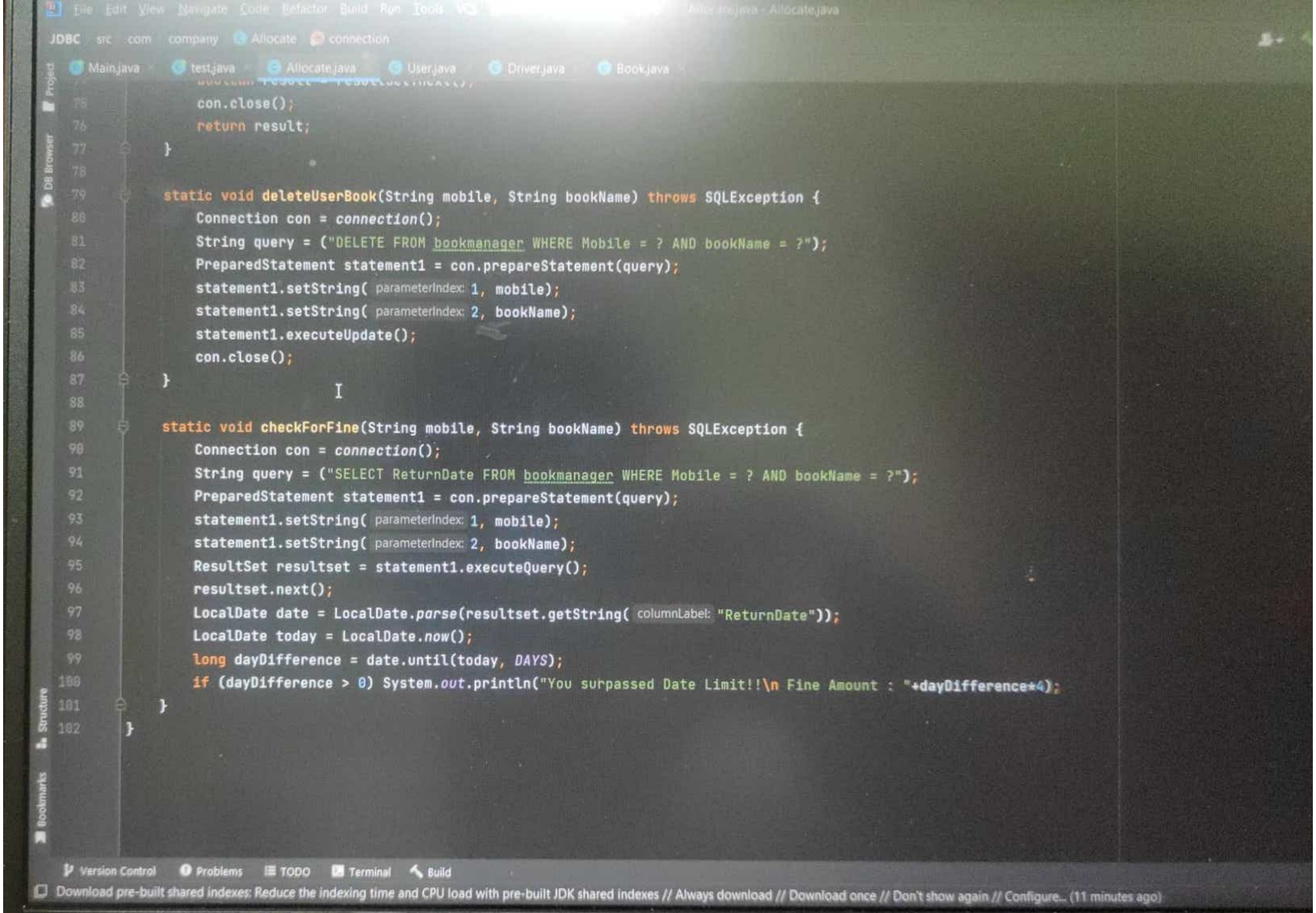

File Edit View Navigate Code Refactor Build Run Tools VCS Window DB Navigator Help
Allocate.java - Allocate.java
JDBC › src › com › company › Allocate › connection
Main.java
test.java
Allocate.java
User.java
Driver.java
Book.java
Main
Project
DB Browser
Structure
Bookmarks
Version Control
Problems
TODO
Terminal
Build
10:65

```java
    static void changeBookCount(String mobile, boolean increment) throws SQLException {
        Connection con = connection();
        String query = "select * from user_details Where Mobile = ?";
        PreparedStatement statement1 = con.prepareStatement(query);
        statement1.setString( parameterIndex: 1, mobile);
        ResultSet resultSet = statement1.executeQuery();
        resultSet.next();

        int curCount;
        if (increment) {
            curCount = resultSet.getInt( columnLabel: "BookUsingCount") + 1;
        } else {
            curCount = resultSet.getInt( columnLabel: "BookUsingCount") - 1;
        }

        String query1 = "update user_details set BookUsingCount =? Where Mobile = ?";
        PreparedStatement statement2 = con.prepareStatement(query1);
        statement2.setInt( parameterIndex: 1, curCount);
        statement2.setString( parameterIndex: 2, mobile);
        statement2.executeUpdate();
        con.close();
        System.out.println("User Book Details Updated Successfully");
    }

    static void insertUserBook(String mobile, String bookName) throws SQLException {
        Connection con = connection();
        String query = ("INSERT INTO bookmanager(Mobile,BookName,ReturnDate) values(?,?,?)");
        LocalDate returnDate = LocalDate.now().plusDays(10);
        PreparedStatement statement1 = con.prepareStatement(query);
        statement1.setString( parameterIndex: 1, mobile);
        statement1.setDate( parameterIndex: 3, Date.valueOf(returnDate));
        statement1.setString( parameterIndex: 2, bookName);
        statement1.executeUpdate();
```

Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built JDK shared indexes // Always download // Download once // Don't show again // Configure... (11 minutes ago)

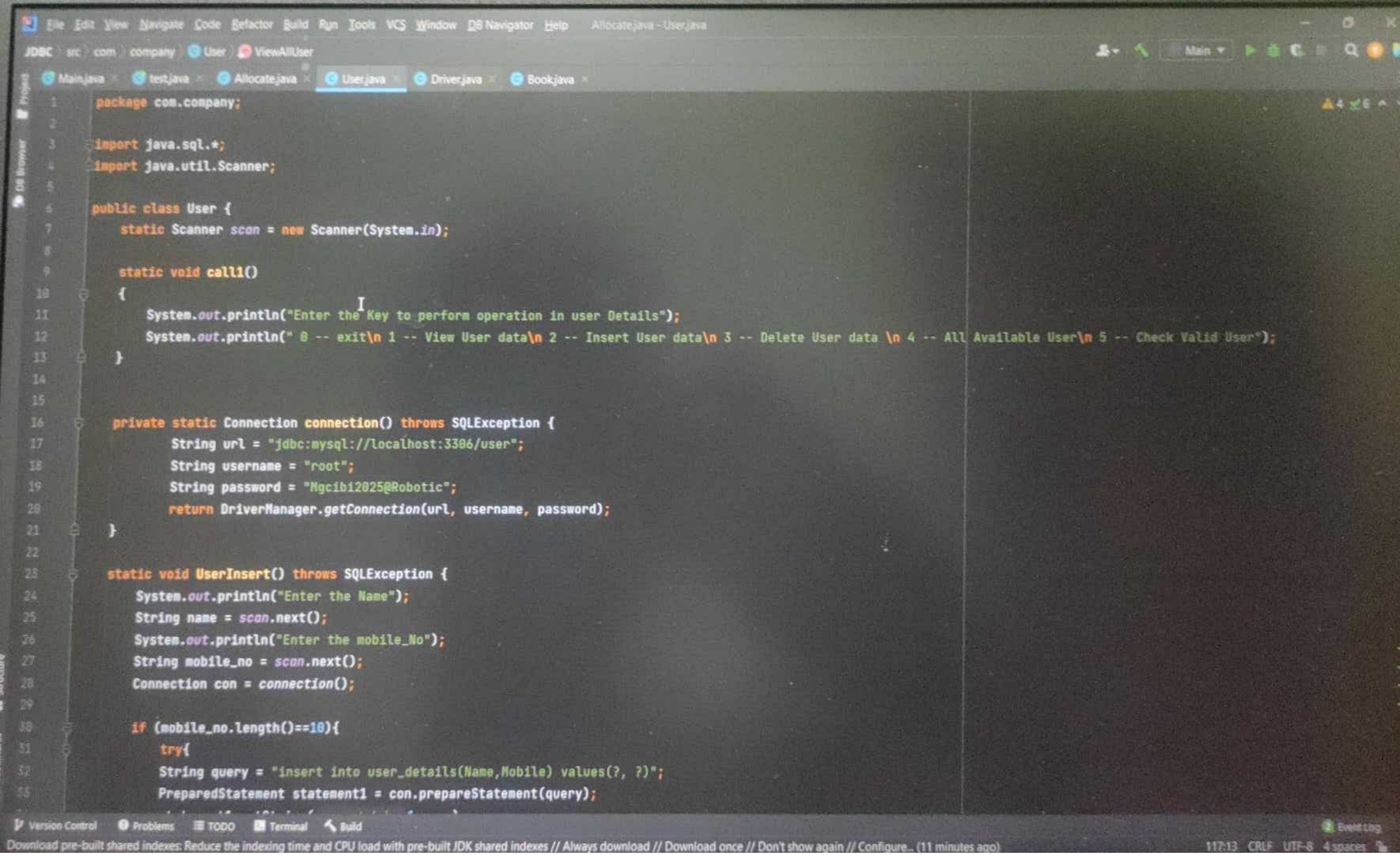

File Edit View Navigate Code Refactor Build Run Tools VCS Window DB Navigator Help Allocate.java - Allocate.java
JDBC src com company Allocate connection
Main.java
test.java
Allocate.java
User.java
Driver.java
Book.java
Main
Project
DB Browser
Structure
Bookmarks
Version Control
Problems
TODO
Terminal
Build

```
static void insertUserBook(String mobile, String bookName) throws SQLException {
    Connection con = connection();
    String query = ("INSERT INTO bookmanager(Mobile,BookName,ReturnDate) values(?,?,?)");
    LocalDate returnDate = LocalDate.now().plusDays(10);
    PreparedStatement statement1 = con.prepareStatement(query);
    statement1.setString( parameterIndex: 1, mobile);
    statement1.setDate( parameterIndex: 3, Date.valueOf(returnDate));
    statement1.setString( parameterIndex: 2, bookName);
    statement1.executeUpdate();
    con.close();
}

static boolean checkBookPresence(String mobile, String bookName) throws SQLException {
    Connection con = connection();
    String query = ("SELECT * FROM bookmanager WHERE Mobile = ? AND BookName = ?");
    PreparedStatement statement1 = con.prepareStatement(query);
    statement1.setString( parameterIndex: 1, mobile);
    statement1.setString( parameterIndex: 2,bookName);
    ResultSet resultSet = statement1.executeQuery();
    boolean result = resultSet.next();
    con.close();
    return result;
}

static void deleteUserBook(String mobile, String bookName) throws SQLException {
    Connection con = connection();
    String query = ("DELETE FROM bookmanager WHERE Mobile = ? AND bookName = ?");
    PreparedStatement statement1 = con.prepareStatement(query);
    statement1.setString( parameterIndex: 1, mobile);
    statement1.setString( parameterIndex: 2, bookName);
    statement1.executeUpdate();
    con.close();
}
```


File Edit View Navigate Code Refactor Build Run Tools VCS
Allocate.java - Allocate.java
JDBC src com company Allocate connection
Main.java test.java Allocate.java User.java Driver.java Book.java
Project
DB Browser
Structure
Bookmarks
Version Control Problems TODO Terminal Build

```
        con.close();
        return result;
    }

    static void deleteUserBook(String mobile, String bookName) throws SQLException {
        Connection con = connection();
        String query = ("DELETE FROM bookmanager WHERE Mobile = ? AND bookName = ?");
        PreparedStatement statement1 = con.prepareStatement(query);
        statement1.setString( parameterIndex: 1, mobile);
        statement1.setString( parameterIndex: 2, bookName);
        statement1.executeUpdate();
        con.close();
    }

    static void checkForFine(String mobile, String bookName) throws SQLException {
        Connection con = connection();
        String query = ("SELECT ReturnDate FROM bookmanager WHERE Mobile = ? AND bookName = ?");
        PreparedStatement statement1 = con.prepareStatement(query);
        statement1.setString( parameterIndex: 1, mobile);
        statement1.setString( parameterIndex: 2, bookName);
        ResultSet resultset = statement1.executeQuery();
        resultset.next();
        LocalDate date = LocalDate.parse(resultset.getString( columnLabel: "ReturnDate"));
        LocalDate today = LocalDate.now();
        long dayDifference = date.until(today, DAYS);
        if (dayDifference > 0) System.out.println("You surpassed Date Limit!!\n Fine Amount : "+dayDifference*4);
    }
}
```

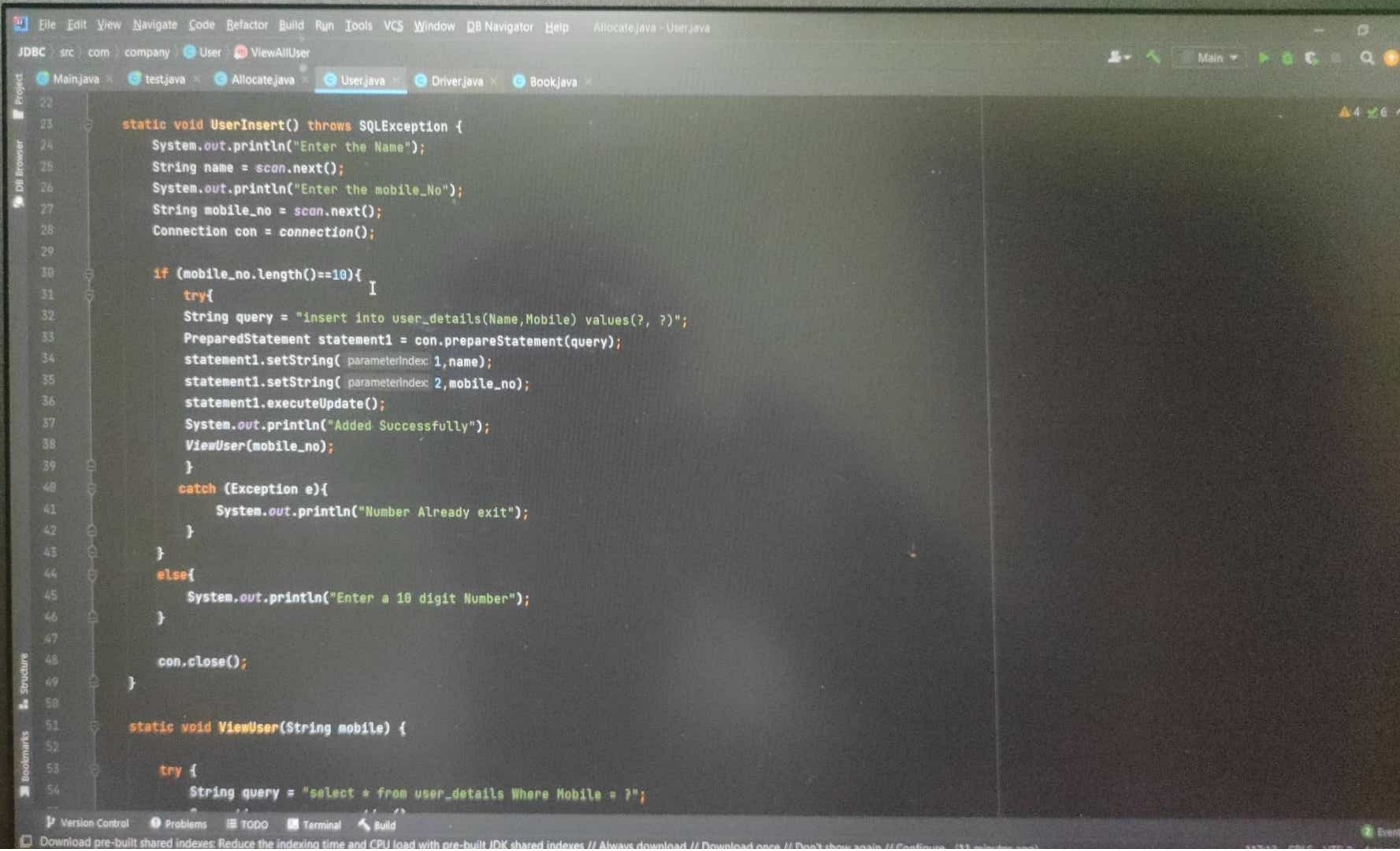
Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built JDK shared indexes // Always download // Download once // Don't show again // Configure... (11 minutes ago)


File Edit View Navigate Code Refactor Build Run Tools VCS Window DB Navigator Help
Allocate.java - User.java
JDBC › src › com › company › User › ViewAllUser
Main
Main.java
test.java
Allocate.java
User.java
Driver.java
Book.java
Project
DB Browser
Version Control
Problems
TODO
Terminal
Build
Event Log

```
package com.company;

import java.sql.*;
import java.util.Scanner;

public class User {
    static Scanner scan = new Scanner(System.in);

    static void call1()
    {
        System.out.println("Enter the Key to perform operation in user Details");
        System.out.println(" 0 -- exit\n 1 -- View User data\n 2 -- Insert User data\n 3 -- Delete User data \n 4 -- All Available User\n 5 -- Check Valid User");
    }


   private static Connection connection() throws SQLException {
           String url = "jdbc:mysql://localhost:3306/user";
           String username = "root";
           String password = "Mgcibi2025@Robotic";
           return DriverManager.getConnection(url, username, password);
   }

  static void UserInsert() throws SQLException {
      System.out.println("Enter the Name");
      String name = scan.next();
      System.out.println("Enter the mobile_No");
      String mobile_no = scan.next();
      Connection con = connection();

      if (mobile_no.length()==10){
          try{
          String query = "insert into user_details(Name,Mobile) values(?, ?)";
          PreparedStatement statement1 = con.prepareStatement(query);
```

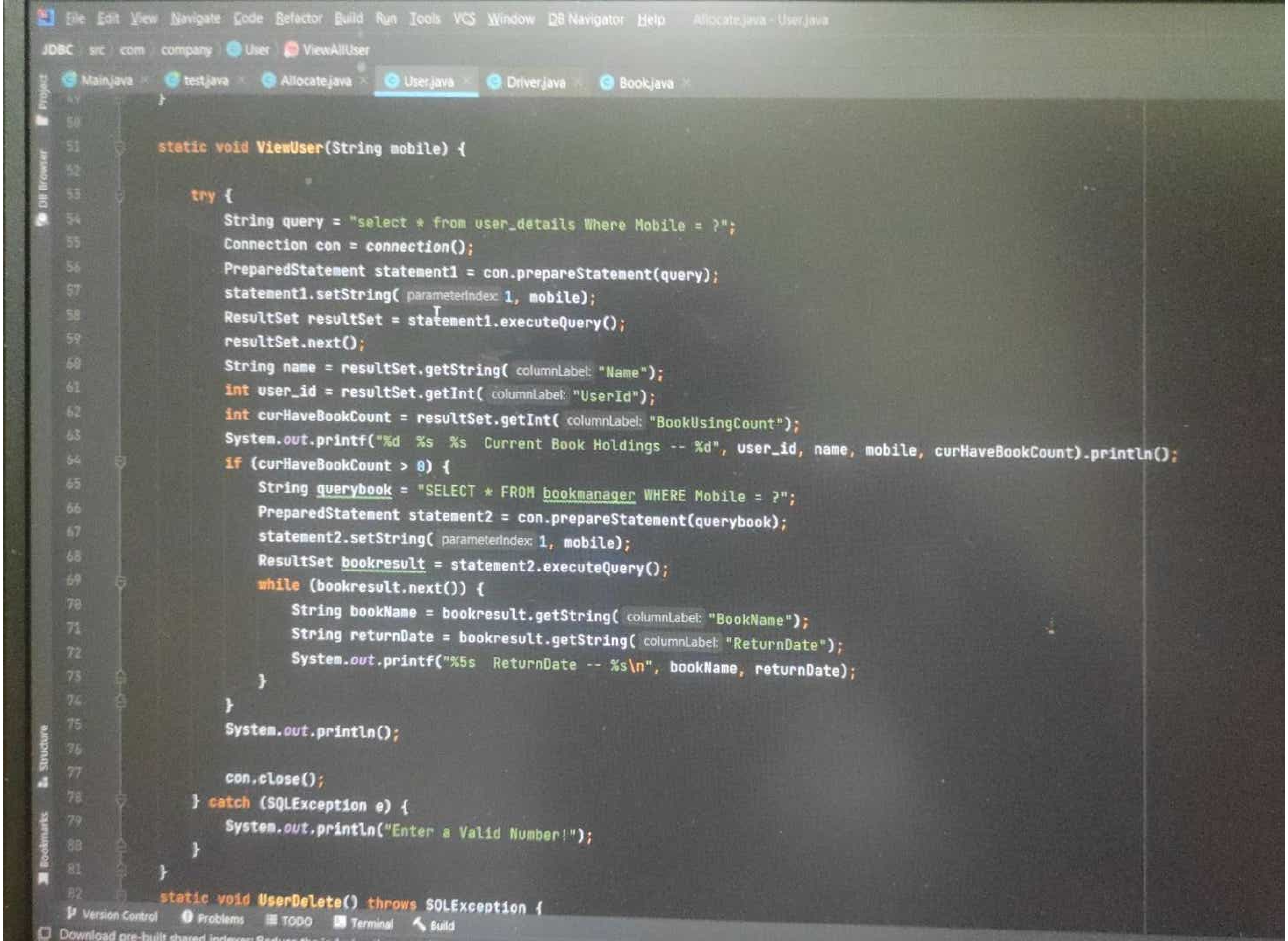
Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built JDK shared indexes // Always download // Download once // Don't show again // Configure... (11 minutes ago)

117:13 CRLF UTF-8 4 spaces

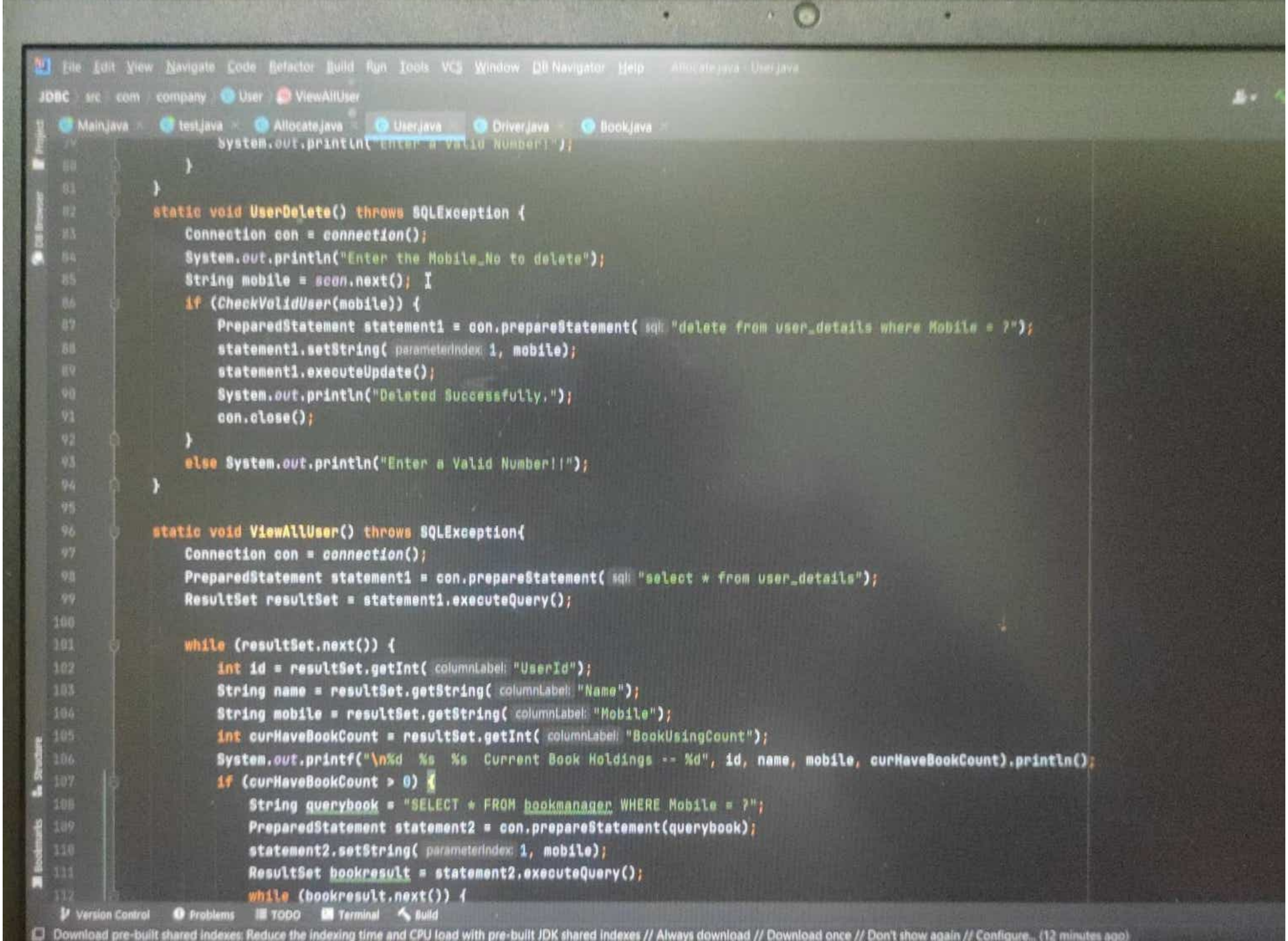

File Edit View Navigate Code Refactor Build Run Tools VCS Window DB Navigator Help Allocate.java - User.java
JDBC src com company User ViewAllUser
Main.java testjava Allocate.java User.java Driver.java Book.java
Main
Project
DB Browser
Structure
Bookmarks
Version Control Problems TODO Terminal Build

```java
    static void UserInsert() throws SQLException {
        System.out.println("Enter the Name");
        String name = scan.next();
        System.out.println("Enter the mobile_No");
        String mobile_no = scan.next();
        Connection con = connection();

        if (mobile_no.length()==10){
            try{
            String query = "insert into user_details(Name,Mobile) values(?, ?)";
            PreparedStatement statement1 = con.prepareStatement(query);
            statement1.setString( parameterIndex: 1,name);
            statement1.setString( parameterIndex: 2,mobile_no);
            statement1.executeUpdate();
            System.out.println("Added Successfully");
            ViewUser(mobile_no);
            }
           catch (Exception e){
               System.out.println("Number Already exit");
            }
        }
        else{
            System.out.println("Enter a 10 digit Number");
        }

        con.close();
    }

    static void ViewUser(String mobile) {

        try {
            String query = "select * from user_details Where Mobile = ?";
```

Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built JDK shared indexes // Always download // Download once // Don't show again // Configure... (11 minutes ago)

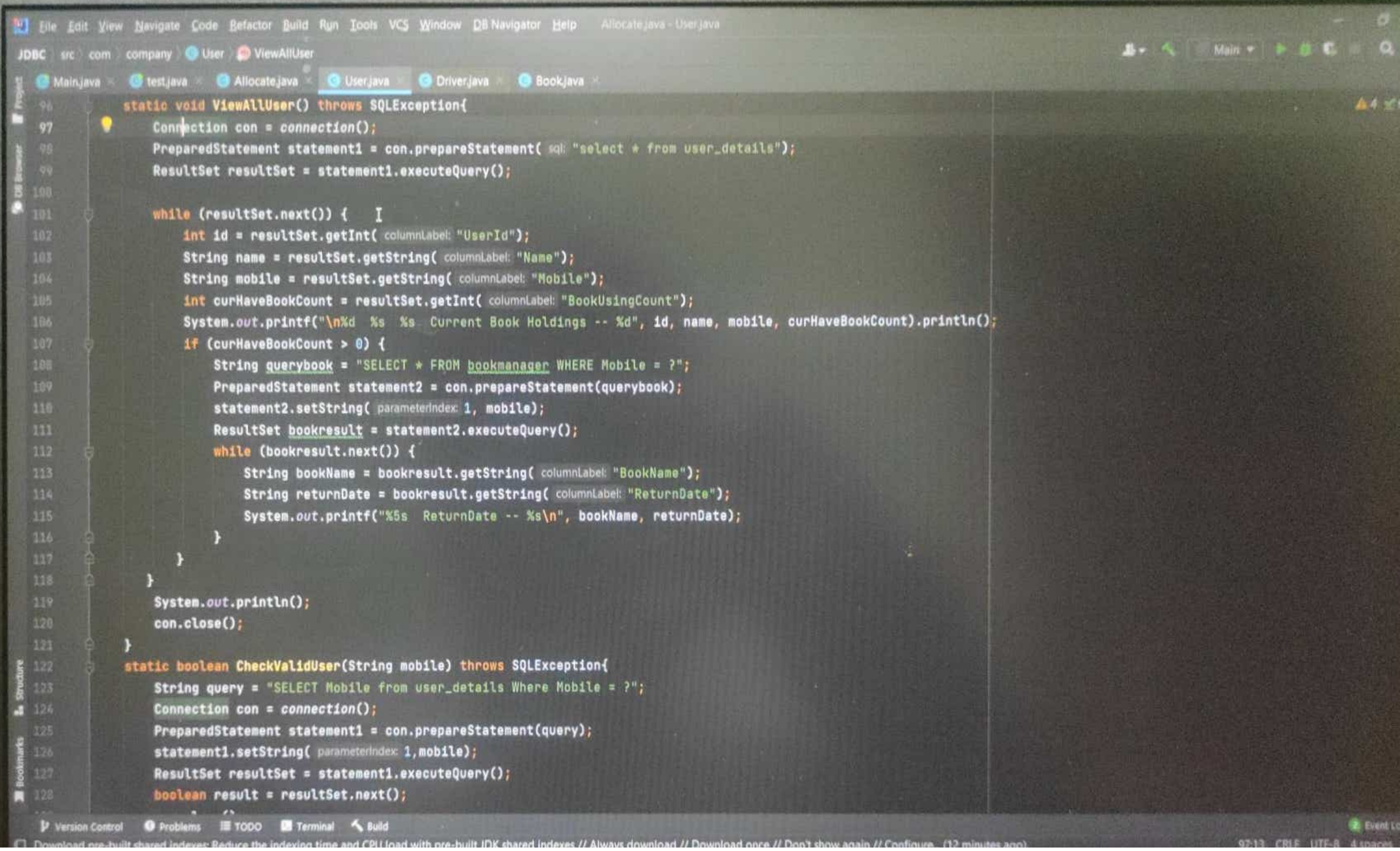

File Edit View Navigate Code Refactor Build Run Tools VCS Window DB Navigator Help Allocate.java - User.java
JDBC src com company User ViewAllUser
Main.java test.java Allocate.java User.java Driver.java Book.java
Project DB Browser Structure Bookmarks
Version Control Problems TODO Terminal Build
Download pre-built shared indexes: Reduce the ind

```java
    }

    static void ViewUser(String mobile) {

        try {
            String query = "select * from user_details Where Mobile = ?";
            Connection con = connection();
            PreparedStatement statement1 = con.prepareStatement(query);
            statement1.setString( parameterIndex: 1, mobile);
            ResultSet resultSet = statement1.executeQuery();
            resultSet.next();
            String name = resultSet.getString( columnLabel: "Name");
            int user_id = resultSet.getInt( columnLabel: "UserId");
            int curHaveBookCount = resultSet.getInt( columnLabel: "BookUsingCount");
            System.out.printf("%d  %s  %s  Current Book Holdings -- %d", user_id, name, mobile, curHaveBookCount).println();
            if (curHaveBookCount > 0) {
                String querybook = "SELECT * FROM bookmanager WHERE Mobile = ?";
                PreparedStatement statement2 = con.prepareStatement(querybook);
                statement2.setString( parameterIndex: 1, mobile);
                ResultSet bookresult = statement2.executeQuery();
                while (bookresult.next()) {
                    String bookName = bookresult.getString( columnLabel: "BookName");
                    String returnDate = bookresult.getString( columnLabel: "ReturnDate");
                    System.out.printf("%5s  ReturnDate -- %s\n", bookName, returnDate);
                }
            }
            System.out.println();

            con.close();
        } catch (SQLException e) {
            System.out.println("Enter a Valid Number!");
        }
    }
    static void UserDelete() throws SQLException {
```


File Edit View Navigate Code Refactor Build Run Tools VCS Window DB Navigator Help Allocate.java - User.java
JDBC src com company User ViewAllUser
Main.java
test.java
Allocate.java
User.java
Driver.java
Book.java
Project
DB Browser
Structure
Bookmarks
Version Control
Problems
TODO
Terminal
Build

```
            System.out.println("Enter a Valid Number!");
        }
    }
    static void UserDelete() throws SQLException {
        Connection con = connection();
        System.out.println("Enter the Mobile_No to delete");
        String mobile = scan.next();
        if (CheckValidUser(mobile)) {
            PreparedStatement statement1 = con.prepareStatement( sql: "delete from user_details where Mobile = ?");
            statement1.setString( parameterIndex: 1, mobile);
            statement1.executeUpdate();
            System.out.println("Deleted Successfully.");
            con.close();
        }
        else System.out.println("Enter a Valid Number!!");
    }

    static void ViewAllUser() throws SQLException{
        Connection con = connection();
        PreparedStatement statement1 = con.prepareStatement( sql: "select * from user_details");
        ResultSet resultSet = statement1.executeQuery();

        while (resultSet.next()) {
            int id = resultSet.getInt( columnLabel: "UserId");
            String name = resultSet.getString( columnLabel: "Name");
            String mobile = resultSet.getString( columnLabel: "Mobile");
            int curHaveBookCount = resultSet.getInt( columnLabel: "BookUsingCount");
            System.out.printf("\n%d   %s   %s  Current Book Holdings -- %d", id, name, mobile, curHaveBookCount).println();
            if (curHaveBookCount > 0) {
                String querybook = "SELECT * FROM bookmanager WHERE Mobile = ?";
                PreparedStatement statement2 = con.prepareStatement(querybook);
                statement2.setString( parameterIndex: 1, mobile);
                ResultSet bookresult = statement2.executeQuery();
                while (bookresult.next()) {
```

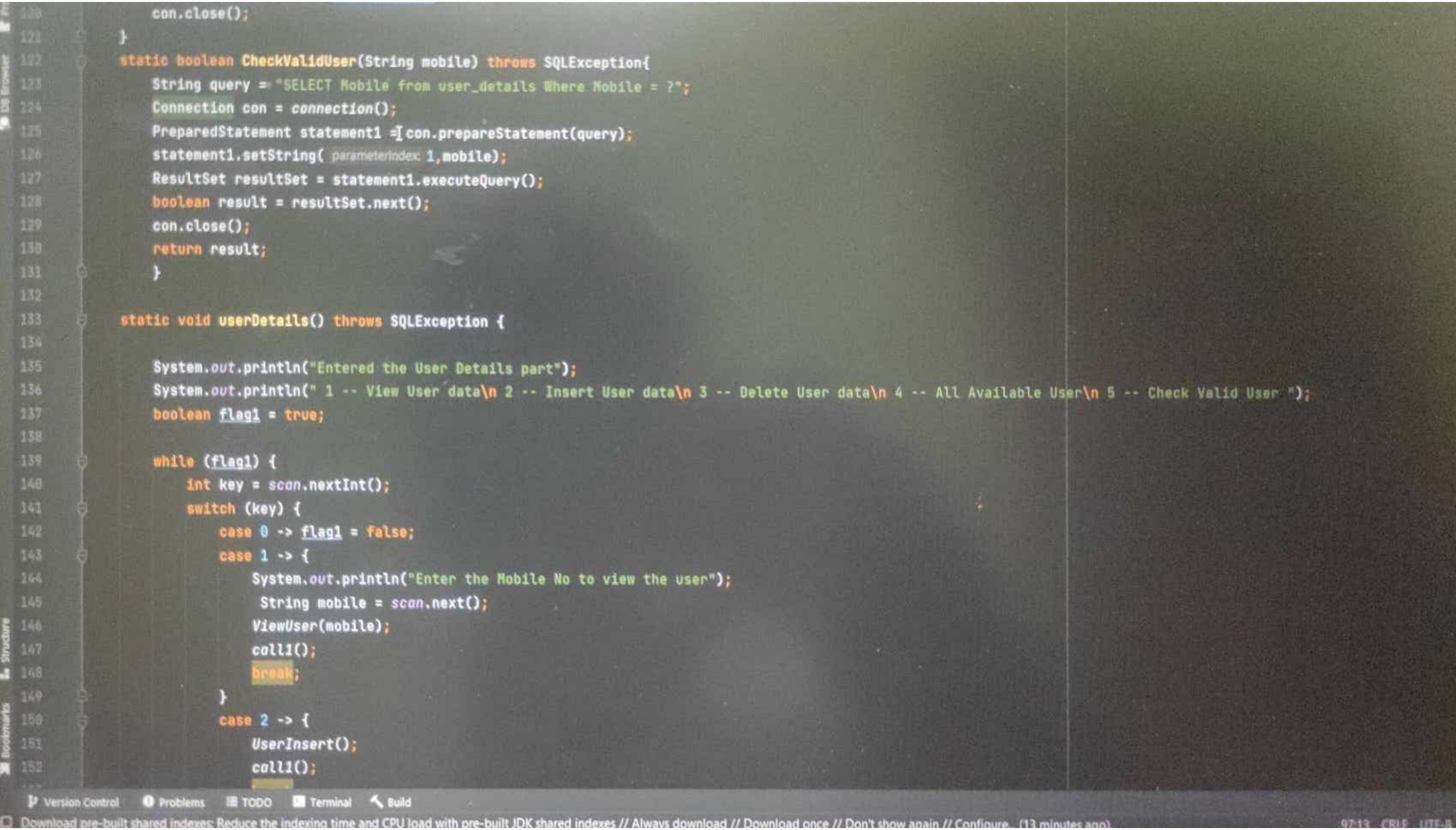
Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built JDK shared indexes // Always download // Download once // Don't show again // Configure... (12 minutes ago)


File Edit View Navigate Code Refactor Build Run Tools VCS Window DB Navigator Help
Allocate.java - User.java
JDBC › src › com › company › User › ViewAllUser
Main.java
test.java
Allocate.java
User.java
Driver.java
Book.java
Main
Project
DB Browser
Structure
Bookmarks
Version Control
Problems
TODO
Terminal
Build
Event Log

```java
static void ViewAllUser() throws SQLException{
    Connection con = connection();
    PreparedStatement statement1 = con.prepareStatement( sql: "select * from user_details");
    ResultSet resultSet = statement1.executeQuery();

    while (resultSet.next()) {
        int id = resultSet.getInt( columnLabel: "UserId");
        String name = resultSet.getString( columnLabel: "Name");
        String mobile = resultSet.getString( columnLabel: "Mobile");
        int curHaveBookCount = resultSet.getInt( columnLabel: "BookUsingCount");
        System.out.printf("\n%d  %s  %s  Current Book Holdings -- %d", id, name, mobile, curHaveBookCount).println();
        if (curHaveBookCount > 0) {
            String querybook = "SELECT * FROM bookmanager WHERE Mobile = ?";
            PreparedStatement statement2 = con.prepareStatement(querybook);
            statement2.setString( parameterIndex: 1, mobile);
            ResultSet bookresult = statement2.executeQuery();
            while (bookresult.next()) {
                String bookName = bookresult.getString( columnLabel: "BookName");
                String returnDate = bookresult.getString( columnLabel: "ReturnDate");
                System.out.printf("%5s  ReturnDate -- %s\n", bookName, returnDate);
            }
        }
    }
    System.out.println();
    con.close();
}
static boolean CheckValidUser(String mobile) throws SQLException{
    String query = "SELECT Mobile from user_details Where Mobile = ?";
    Connection con = connection();
    PreparedStatement statement1 = con.prepareStatement(query);
    statement1.setString( parameterIndex: 1,mobile);
    ResultSet resultSet = statement1.executeQuery();
    boolean result = resultSet.next();
```

Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built JDK shared indexes // Always download // Download once // Don't show again // Configure... (12 minutes ago)

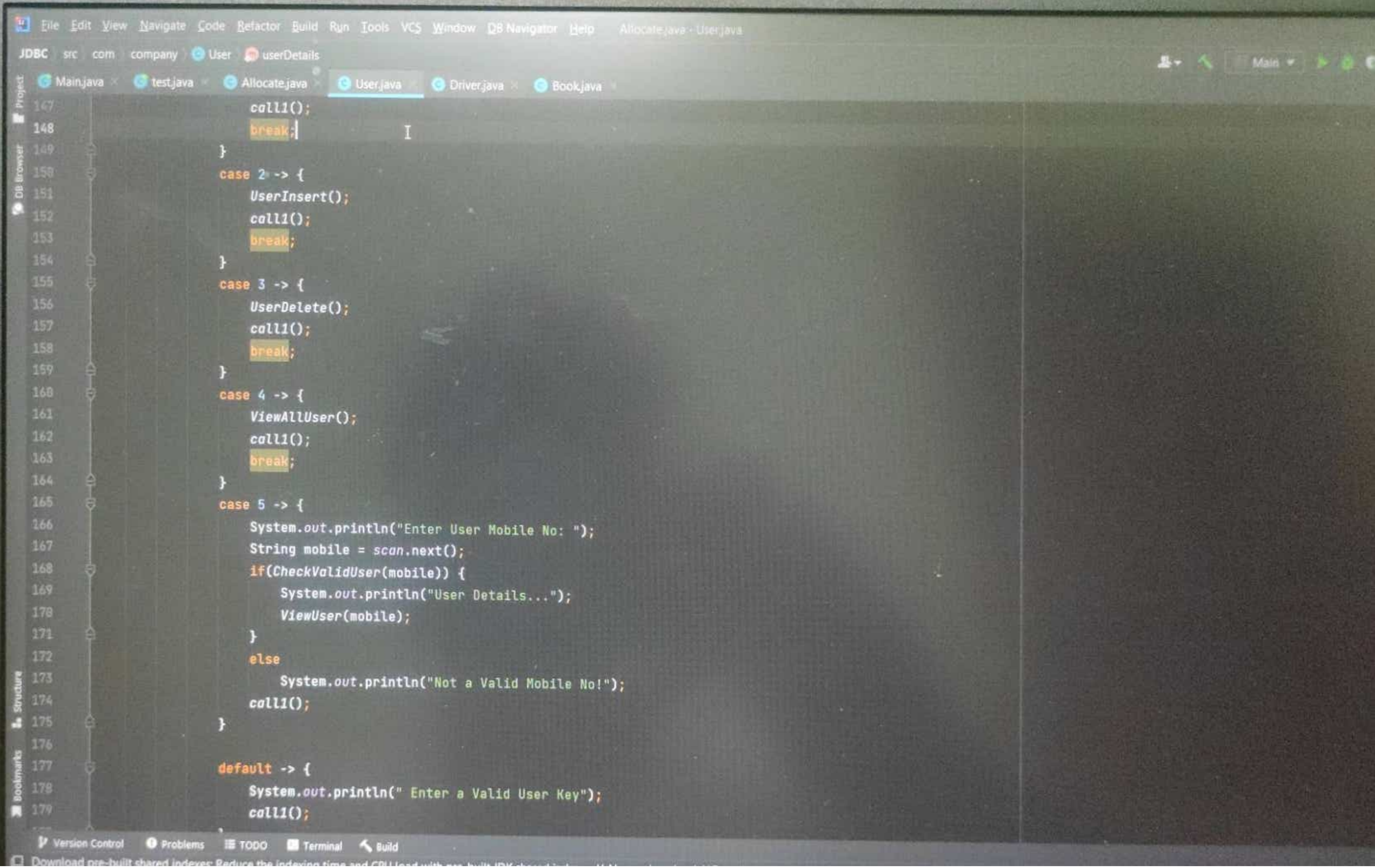
97:13 CRLF UTF-8 4 spaces



```
        con.close();
    }
    static boolean CheckValidUser(String mobile) throws SQLException{
        String query = "SELECT Mobile from user_details Where Mobile = ?";
        Connection con = connection();
        PreparedStatement statement1 = con.prepareStatement(query);
        statement1.setString( parameterIndex: 1,mobile);
        ResultSet resultSet = statement1.executeQuery();
        boolean result = resultSet.next();
        con.close();
        return result;
        }

    static void userDetails() throws SQLException {

        System.out.println("Entered the User Details part");
        System.out.println(" 1 -- View User data\n 2 -- Insert User data\n 3 -- Delete User data\n 4 -- All Available User\n 5 -- Check Valid User ");
        boolean flag1 = true;

        while (flag1) {
            int key = scan.nextInt();
            switch (key) {
                case 0 -> flag1 = false;
                case 1 -> {
                    System.out.println("Enter the Mobile No to view the user");
                     String mobile = scan.next();
                    ViewUser(mobile);
                    call1();
                    break;
                }
                case 2 -> {
                    UserInsert();
                    call1();
```

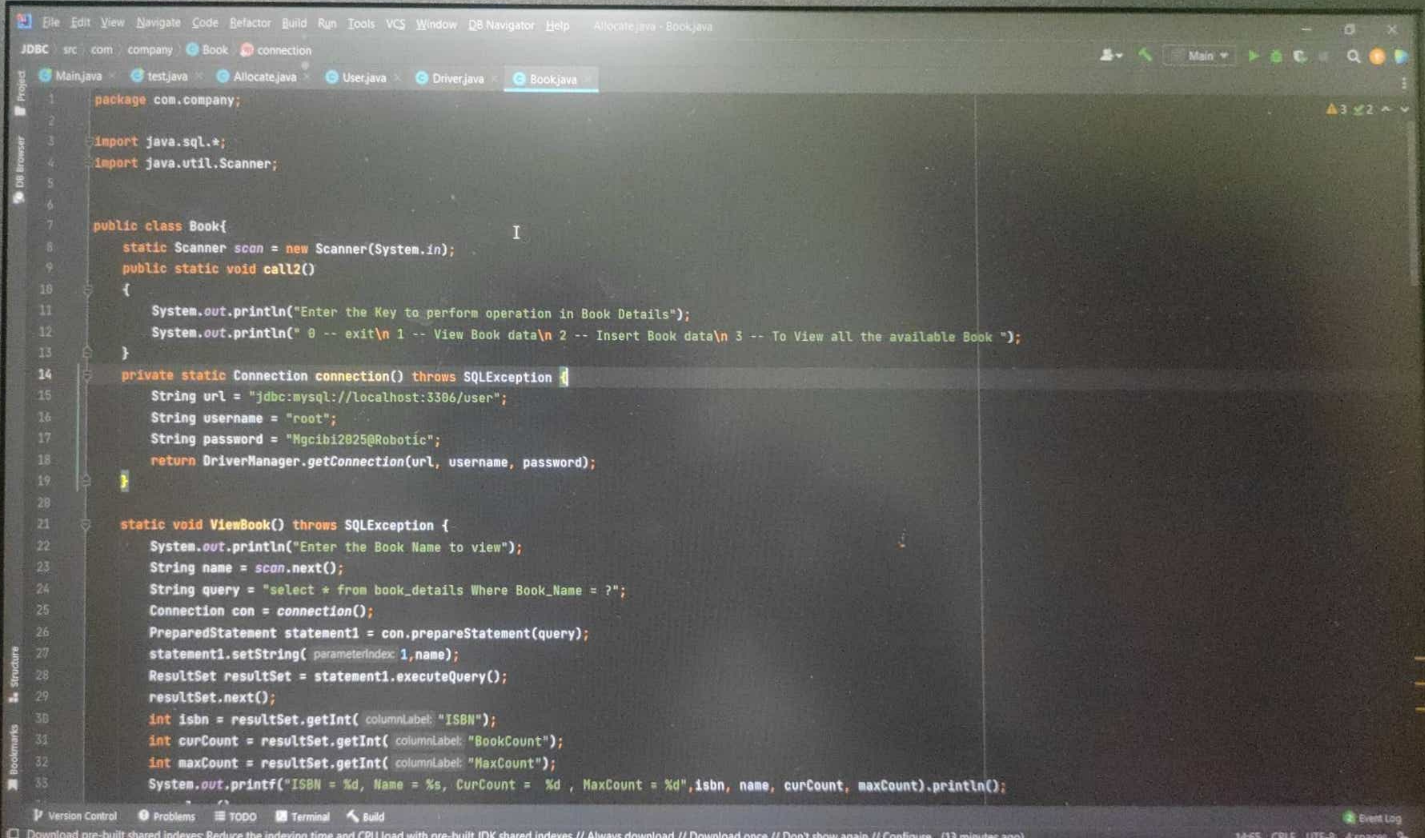
Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built JDK shared indexes // Always download // Download once // Don't show again // Configure... (13 minutes ago)

97:13 CRLF UTF-8

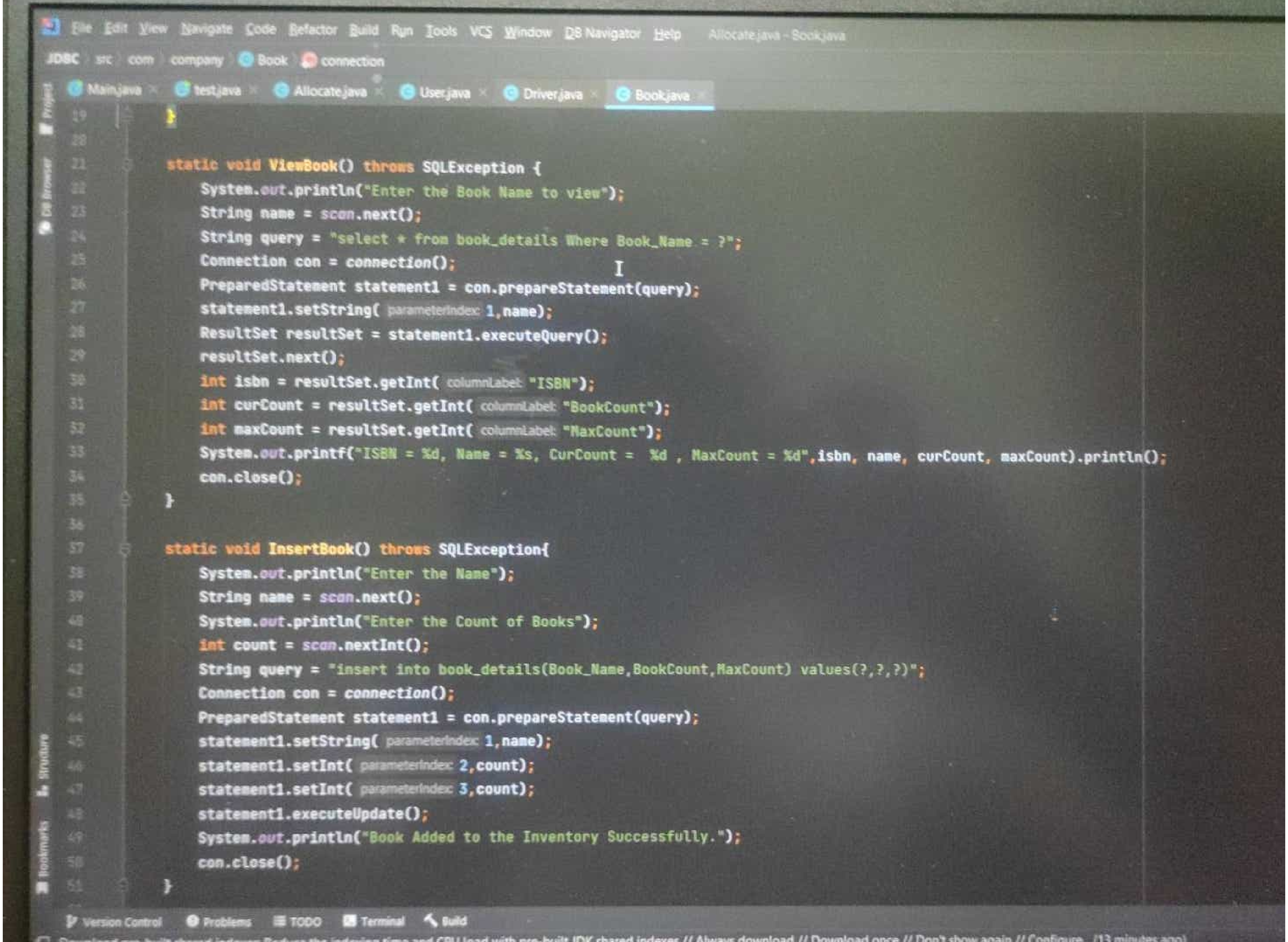

File Edit View Navigate Code Refactor Build Run Tools VCS Window DB Navigator Help
Allocate.java - User.java
JDBC src com company User userDetails
Main
Main.java
test.java
Allocate.java
User.java
Driver.java
Book.java
Project
DB Browser
Structure
Bookmarks
Version Control
Problems
TODO
Terminal
Build

```
        call1();
        break;
    }
    case 2 -> {
        UserInsert();
        call1();
        break;
    }
    case 3 -> {
        UserDelete();
        call1();
        break;
    }
    case 4 -> {
        ViewAllUser();
        call1();
        break;
    }
    case 5 -> {
        System.out.println("Enter User Mobile No: ");
        String mobile = scan.next();
        if(CheckValidUser(mobile)) {
            System.out.println("User Details...");
            ViewUser(mobile);
        }
        else
            System.out.println("Not a Valid Mobile No!");
        call1();
    }

    default -> {
        System.out.println(" Enter a Valid User Key");
        call1();
```

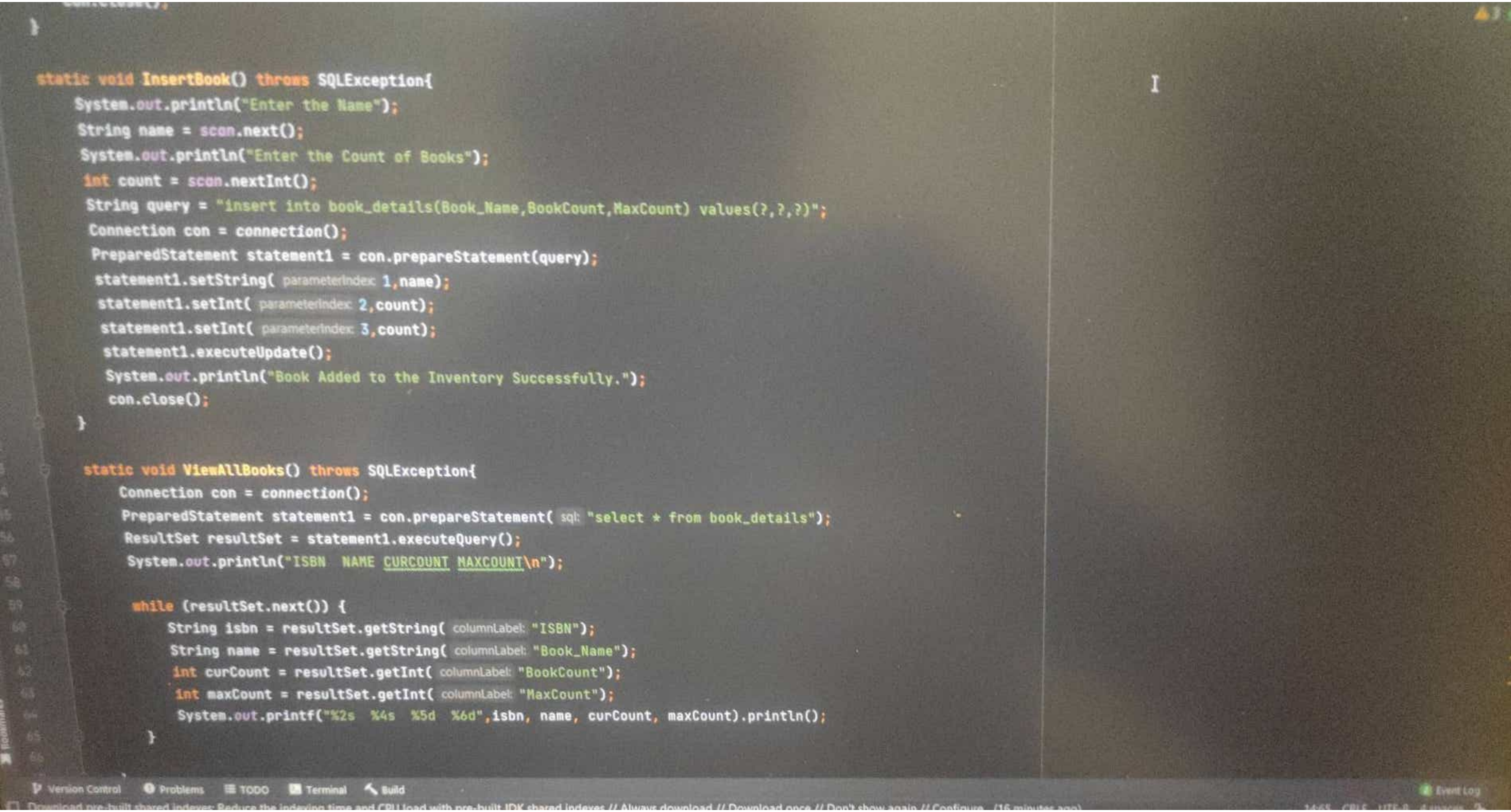
Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built JDK shared in


File Edit View Navigate Code Refactor Build Run Tools VCS Window DB Navigator Help Allocate.java - Book.java
JDBC › src › com › company › Book › connection
Main.java
test.java
Allocate.java
User.java
Driver.java
Book.java
Main
Project
DB Browser
Structure
Bookmarks
Version Control
Problems
TODO
Terminal
Build
Event Log

```java
package com.company;

import java.sql.*;
import java.util.Scanner;


public class Book{
    static Scanner scan = new Scanner(System.in);
    public static void call2()
    {
        System.out.println("Enter the Key to perform operation in Book Details");
        System.out.println(" 0 -- exit\n 1 -- View Book data\n 2 -- Insert Book data\n 3 -- To View all the available Book ");
    }
    private static Connection connection() throws SQLException {
        String url = "jdbc:mysql://localhost:3306/user";
        String username = "root";
        String password = "Mgcibi2025@Robotic";
        return DriverManager.getConnection(url, username, password);
    }

    static void ViewBook() throws SQLException {
        System.out.println("Enter the Book Name to view");
        String name = scan.next();
        String query = "select * from book_details Where Book_Name = ?";
        Connection con = connection();
        PreparedStatement statement1 = con.prepareStatement(query);
        statement1.setString( parameterIndex: 1,name);
        ResultSet resultSet = statement1.executeQuery();
        resultSet.next();
        int isbn = resultSet.getInt( columnLabel: "ISBN");
        int curCount = resultSet.getInt( columnLabel: "BookCount");
        int maxCount = resultSet.getInt( columnLabel: "MaxCount");
        System.out.printf("ISBN = %d, Name = %s, CurCount =  %d , MaxCount = %d",isbn, name, curCount, maxCount).println();
```

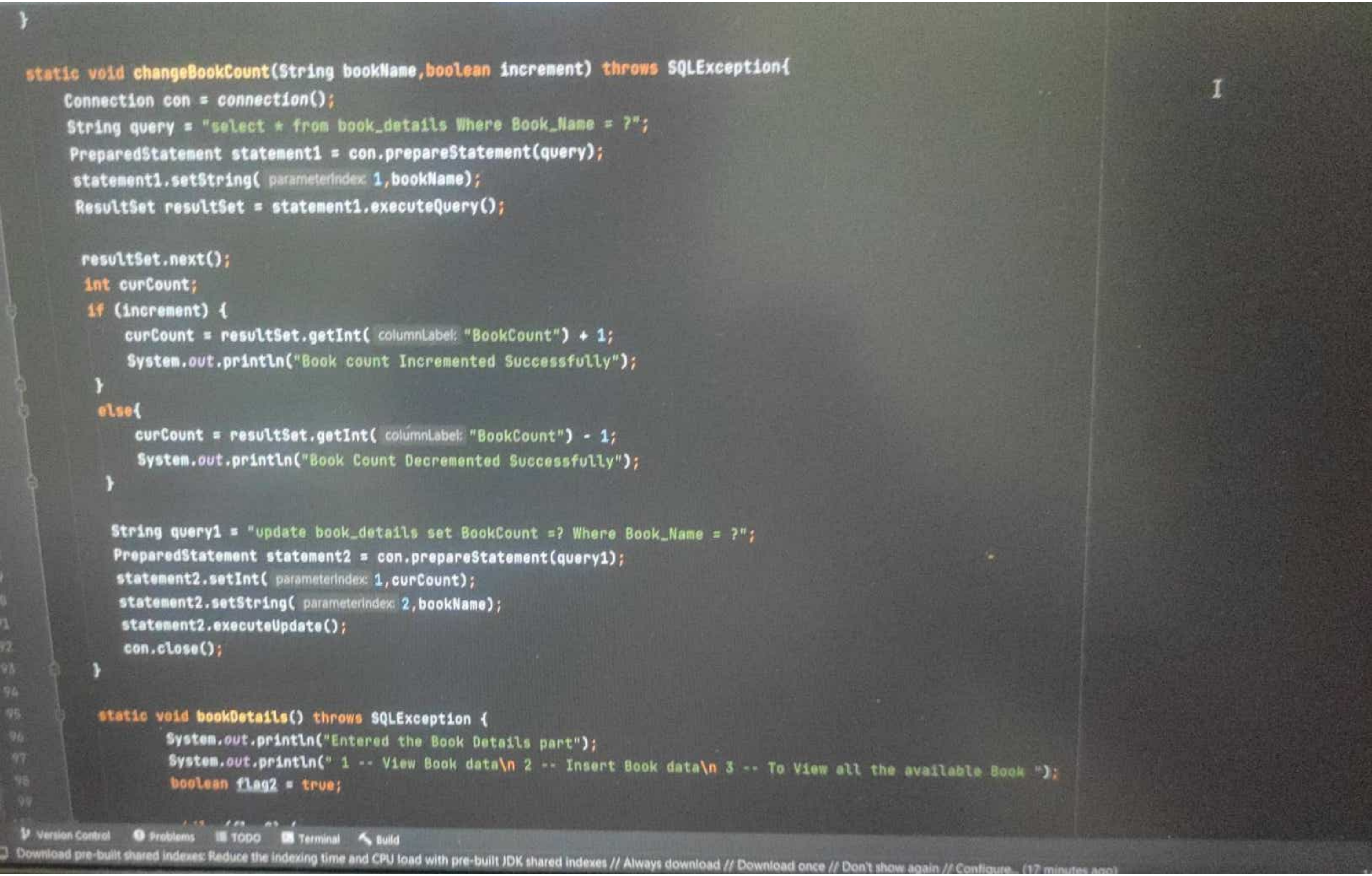

File Edit View Navigate Code Refactor Build Run Tools VCS Window DB Navigator Help
Allocate.java - Book.java
JDBC › src › com › company › Book › connection
Main.java
test.java
Allocate.java
User.java
Driver.java
Book.java
Project
DB Browser
Structure
Bookmarks
Version Control
Problems
TODO
Terminal
Build

```java
    }

    static void ViewBook() throws SQLException {
        System.out.println("Enter the Book Name to view");
        String name = scan.next();
        String query = "select * from book_details Where Book_Name = ?";
        Connection con = connection();
        PreparedStatement statement1 = con.prepareStatement(query);
        statement1.setString( parameterIndex: 1,name);
        ResultSet resultSet = statement1.executeQuery();
        resultSet.next();
        int isbn = resultSet.getInt( columnLabel: "ISBN");
        int curCount = resultSet.getInt( columnLabel: "BookCount");
        int maxCount = resultSet.getInt( columnLabel: "MaxCount");
        System.out.printf("ISBN = %d, Name = %s, CurCount =  %d , MaxCount = %d",isbn, name, curCount, maxCount).println();
        con.close();
    }

    static void InsertBook() throws SQLException{
        System.out.println("Enter the Name");
        String name = scan.next();
        System.out.println("Enter the Count of Books");
        int count = scan.nextInt();
        String query = "insert into book_details(Book_Name,BookCount,MaxCount) values(?,?,?)";
        Connection con = connection();
        PreparedStatement statement1 = con.prepareStatement(query);
        statement1.setString( parameterIndex: 1,name);
        statement1.setInt( parameterIndex: 2,count);
        statement1.setInt( parameterIndex: 3,count);
        statement1.executeUpdate();
        System.out.println("Book Added to the Inventory Successfully.");
        con.close();
    }
```



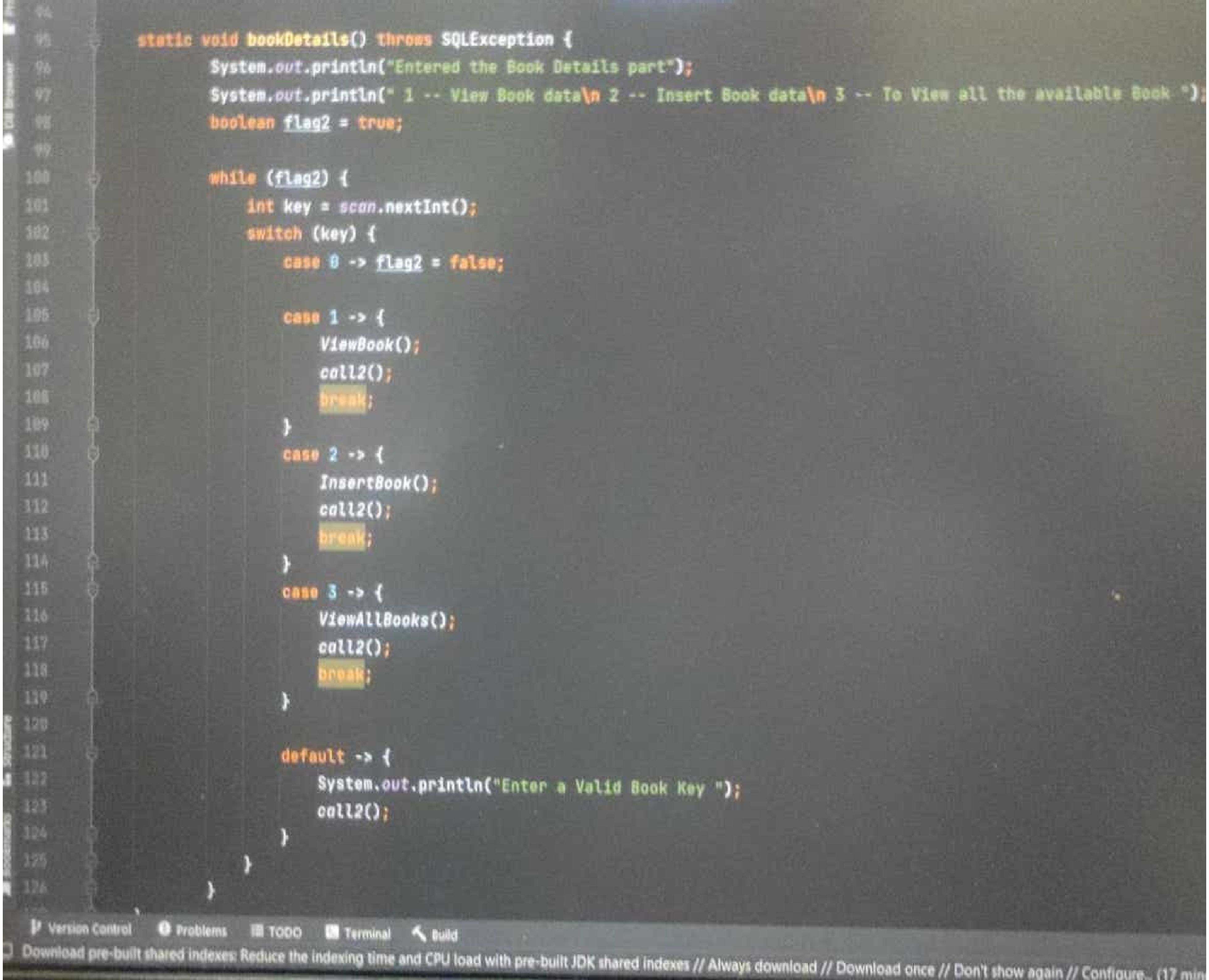
```
    con.close();
}

static void InsertBook() throws SQLException{
    System.out.println("Enter the Name");
    String name = scan.next();
    System.out.println("Enter the Count of Books");
    int count = scan.nextInt();
    String query = "insert into book_details(Book_Name,BookCount,MaxCount) values(?,?,?)";
    Connection con = connection();
    PreparedStatement statement1 = con.prepareStatement(query);
    statement1.setString( parameterIndex: 1,name);
    statement1.setInt( parameterIndex: 2,count);
    statement1.setInt( parameterIndex: 3,count);
    statement1.executeUpdate();
    System.out.println("Book Added to the Inventory Successfully.");
    con.close();
}

static void ViewAllBooks() throws SQLException{
    Connection con = connection();
    PreparedStatement statement1 = con.prepareStatement( sql: "select * from book_details");
    ResultSet resultSet = statement1.executeQuery();
    System.out.println("ISBN  NAME CURCOUNT MAXCOUNT\n");

    while (resultSet.next()) {
        String isbn = resultSet.getString( columnLabel: "ISBN");
        String name = resultSet.getString( columnLabel: "Book_Name");
        int curCount = resultSet.getInt( columnLabel: "BookCount");
        int maxCount = resultSet.getInt( columnLabel: "MaxCount");
        System.out.printf("%2s  %4s  %5d  %6d",isbn, name, curCount, maxCount).println();
    }
```

Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built JDK shared indexes // Always download // Download once // Don't show again // Configure... (16 minutes ago)



```
}

static void changeBookCount(String bookName,boolean increment) throws SQLException{
    Connection con = connection();
    String query = "select * from book_details Where Book_Name = ?";
    PreparedStatement statement1 = con.prepareStatement(query);
    statement1.setString( parameterIndex: 1,bookName);
    ResultSet resultSet = statement1.executeQuery();

    resultSet.next();
    int curCount;
    if (increment) {
        curCount = resultSet.getInt( columnLabel: "BookCount") + 1;
        System.out.println("Book count Incremented Successfully");
    }
    else{
        curCount = resultSet.getInt( columnLabel: "BookCount") - 1;
        System.out.println("Book Count Decremented Successfully");
    }

    String query1 = "update book_details set BookCount =? Where Book_Name = ?";
    PreparedStatement statement2 = con.prepareStatement(query1);
    statement2.setInt( parameterIndex: 1,curCount);
    statement2.setString( parameterIndex: 2,bookName);
    statement2.executeUpdate();
    con.close();
}

static void bookDetails() throws SQLException {
        System.out.println("Entered the Book Details part");
        System.out.println(" 1 -- View Book data\n 2 -- Insert Book data\n 3 -- To View all the available Book ");
        boolean flag2 = true;
```

Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built JDK shared indexes // Always download // Download once // Don't show again // Configure... (17 minutes ago)



```
    static void bookDetails() throws SQLException {
        System.out.println("Entered the Book Details part");
        System.out.println(" 1 -- View Book data\n 2 -- Insert Book data\n 3 -- To View all the available Book ");
        boolean flag2 = true;

        while (flag2) {
            int key = scan.nextInt();
            switch (key) {
                case 0 -> flag2 = false;

                case 1 -> {
                    ViewBook();
                    call2();
                    break;
                }
                case 2 -> {
                    InsertBook();
                    call2();
                    break;
                }
                case 3 -> {
                    ViewAllBooks();
                    call2();
                    break;
                }

                default -> {
                    System.out.println("Enter a Valid Book Key ");
                    call2();
                }
            }
        }
```

Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built JDK shared indexes // Always download // Download once // Don't show again // Configure... (17 min