FAQ

#Architecture

Q.1 : Which architecture is used by project 4?

Ans. MVC

Q.2 : What is advantage of MVC architecture ?

Ans. **Advantages of MVC**

1) **Faster development process:** MVC supports rapid and parallel development. With MVC, one programmer can work on the view while other can work on the controller to create business logic of the web application. The application developed using MVC can be three times faster than application developed using other development patterns.
2) **Ability to provide multiple views:** In the MVC Model, you can create multiple views for a model. Code duplication is very limited in MVC because it separates data and business logic from the display.
3) **Support for asynchronous technique:** MVC also supports asynchronous technique, which helps developers to develop an application that loads very fast.
4) **Modification does not affect the entire model**: Modification does not affect the entire model because model part does not depend on the views part. Therefore, any changes in the Model will not affect the entire architecture.
5) **MVC model returns the data without formatting:** MVC pattern returns data without applying any formatting so the same components can be used and called for use with any interface.
6) **SEO friendly Development platform:** Using this platform, it is very easy to develop SEO-friendly URLs to generate more visits from a specific application.

Disadvantages of MVC

1) Increased complexity
2) Inefficiency of data access in view
3) Difficulty of using MVC with modern user interface.
4) Need multiple programmers
5) Knowledge on multiple technologies is required.
6) Developer have knowledge of client side code and html code.

Q.3 : What are the responsibilities of Model, View and Controller components?

Ans.

The Model

Model contains persistent data and the business methods to perform business operations on this data and store the data into database. Ideally one Model represents one RDBMS table. One instance of a Model represents one record of a Table. Model contains business logics, data access logics, integration logics to perform business operations and manipulate database.

The View

View components are responsible to render the graphical user interface. User can enter input data using UI input forms or can see application data as different reports.
View contains presentation logics and display data from Model object. If Model data will be changed view will be changed.
View gets its data from Controller to display at user interface as Model object.

View submits its data to its Controller on a user action. A user action is a button click or a menu item selection from the user interface.
Ideally one View has one Controller. When a large data record is to be displayed in groups at UI then multiple Views will have single Controller.
Web applications implement View by JSPs, Desktop applications implement View by Swing/AWT Frame classes.

The Controller

Controllers contain navigation logics and are responsible to perform business operation submitted by View with the help of Model.
Navigation logics are used to determine next View to be displayed after successful or unsuccessful execution of business operation.
The View submits data to the Controller when user clicks on a button or clicks on a menu item from UI. Controller populates submitted data into Model and calls business method to perform business operation. After performing business operation Controller forward control to the next View.
Web applications implement Controller using Servlets. Desktop applications implement Controller using listener classes.

Q.4 : How many type of logic are there?

Ans. Application logics can be categorized into five categories
1. Presentation Logics: This is the set of control statements that decide look and feel of an application's user interface. It is also called View Logic. JSP pages and Swing/AWT Frame classes represent view logics.
2. Control Logics: This is the set of control statements that decide navigation of user interface falls under this category. It will conditionally decide which next page will be displayed to User. These are also known as navigation logic.
3. Business Logics: This is the set of control statements that perform business operations. Operation such as fund transfer, change password, book order, and payment are the examples of business operations.
4. Data Access Logics: This is the set of control statements which perform database manipulation are called Data Access Logics. JDBC or SQL calls to manipulate data into the database are called data access logics.
5. Integration Logics: This is the set of control statements that integrate application with another applications or servers. For example SMS server, FTP server, Email server integration code is part of these logics.

Q.5 : Which logic will you write in different MVC components?

Ans . View : Presentation Logics
Controller : Control Logics
Model : Business Logics, Data Access Logics, Integration Logics.

#Utility Classes

##DCP

Q.6: How do you write DCP? What are the advantages of DCP?

Ans. We write DCP:
ideally an application needs only one instance of DCP.
The class which have only one instance in their lifetime are called singleton classes.
We need only one instance of DCP in our application thus will make our class a singleton class that will contain DCP object.

There are 4 steps to make a singleton class:

1. Make class final so that child can't be creatred for single class.

2. Make default constructor private so that no one other class can instantiate single class.

3. Declare a static variable of self type in single class, static variables have only one copy in their lifetime.

4. Make a getInstance() static method in singleton class that will return instance of same single class.

Advantages of DCP:

1. Connection reusability and limit of maximum connection.

2. It shares pool connections among multiple users.

3. It avoids creation and destruction of connection again & again that'll increase the perfomance of application.

Q.7: Which library have you used for DCP?

Ans. C3P0 we've used.

Q.8: Why did you use c3p0 library?

Ans. Used and recommended for production (live) application.

Q.9: What is maximum connection pool size and minimum connection pool size?

Ans. Minimum number of connections:

```
datasource.cpds.setMinPoolSize(new Integer((String) rb.getString("minPoolSize"))); //  size 10
```

Maximum number of connections:

```
datasource.cpds.setMaxPoolSize(new Integer((String) rb.getString("maxPoolSize"))); // size 100
```

Q.10: Which design patterns are used by JDBCDataSource class.

Ans. Singleton Design pattern.

Factory Design Pattern.

##Property Reader

Q.11: How do you externalize application configuration parameters?

Ans. By writing the parameters in a text file contain multiple key-value pairs.

Q.12 : What is the extension of your configuration file?

Ans. .properties

Q.13: How do you read properties files?

Ans. By using ResourceBundle object.

Q.14: Which configuration parameters will you keep in properties file?

Ans. Those configuration parameters which removes hard coded strings from java source code.

##Email

Q.15: Which API will you use to send emails?

Ans. Java Mail API

Q.16: How do you attach documents with an email?

Ans. Multipart mp = new MimeMultipart();
MimeBodyPart file = new MimeBodyPart();
file.attachFile();
mp.addBodyPart(file);

Q.17 : What is advantage of EmailUtility class?

Ans. Email Utility provides Email Services

Q.18 : Which version of JavaMail API are you using?

Ans. javax.mail.1.4.3

Q.19 : Which design pattern is followed by EmailBuilder?

Ans. Builder design pattern(making complex object by using simple object by step by step approch).

Q.20:Which classes of JavaMail API will you use for sending an email?

Ans. Transport
Message
Mimemessage
InternetAddress
PasswordAuthenticate
Session

Q.21: Which properties will you set to configure JavaMail session?

Ans. Email Server Parameters
smtp.server=smtp.gmail.com
smtp.port=465
email.login=webmaster@sunrays.co.in
email.pwd=PA$$1234

##DataValidator

Q.22: How do you perform input validation in your application?

Ans. By overriding validate() of BaseCtl class.

Q.23: What is the significance of DataValidator class?

Ans. This class validates input data

Q.24: Which are the different methods in DataValidator class?

Ans. isSelect()
isMarks()
isRollNo()

isGender()
isNull()
isExamDate()
isTime()
isNotNull()
isInteger()
isAlpha()
isLong()
isMobileNo()
isEmail()
isDate()
main()

Q.25: How do you validate an email id?
Ans. String emailreg = "^[_A-Za-z0-9-]+(\\.[_A-Za-z0-9-]+)*@[A-Za-z0-9]+(\\.[A-Za-z0-9]+)*(\\.[A-Za-z]{2,})$";

##DataUtility

Q.26: How do you convert request parameters to java types?
Ans. Using DataUtility class of util pakage.

Q.27: What is the significance of DataUtility class?
Ans. Data Utility class to format data from one format to another.

Q.28: Which are the different methods in DataUtility class?
Ans. String getString(String val)
String getStringData(Object val)
int getInt(String val)
long getLong(String val)
Date getDate(String val)
String getDateString(Date date)
Date getDate(Date date, int day)
Timestamp getTimestamp(String val)
Timestamp getTimestamp(long l)
Timestamp getCurrentTimestamp()
long getTimestamp(Timestamp tm)
main(String[] args)

Q.29: How do you format a date?
Ans. By creating SimpleDateFormat object in DataUtility class. E.g.

```
/**
         * Application Date Format
         */
        public static final String APP_DATE_FORMAT = "MM/dd/yyyy";

        public static final String APP_TIME_FORMAT = "MM/dd/yyyy HH:mm:ss";

        /**
```

```
 * Date formatter
 */
private static final SimpleDateFormat formatter = new SimpleDateFormat(
                APP_DATE_FORMAT);

private static final SimpleDateFormat timeFormatter = new SimpleDateFormat(
                APP_TIME_FORMAT);
```

##HTMLUtility

Q.30: Why do you use HTMLUtility class?

Ans. HTML Utility class to produce HTML contents like Dropdown List.

Q.31: What is the significance of HTMLUtility class?

Ans.

Q.32: How do you display drop down list using HTMLUtility?

Ans. By overloaded getList().

##ServletUtility

Q.33: Why do you use ServletUtility class?

Ans. This class provides utility operation for Servlet container like forward, redirect, handle generic exception, manage success and error message, manage default Bean and List, manage pagination parameters.

Q.34: What is the significance of ServletUtility class?

Ans.

#Bean

Q.35: Why do you use BaseBean?

Ans. Parent class of all Beans in application. It contains generic attributes.

Q.36: Why do you use ID as non business primary key?

Ans. Because we doesn't provide any relation between ID and other attributes and ID doesn't contain any business related data.

Q.37: Why does BaseBean have createdBy, createdDate, modifiedBy and modifiedDate attributes?

Ans. BaseBean is the parent Bean and these are common attributes for all Bean classes.

Q.38: Why did you implemented DropdownListBean interface

Ans. It contains two methods getkey() and getvalue() and both methods are overridden by all bean classes.

#Controllers

Q.39: What type of logic are contained by a controller?
Ans. Navigation Login (Controlled Logic)

Q.40: Why do you use BaseCtl class?
Ans. Base controller class of project. It contain (1) Generic operations (2) Generic constants (3) Generic work flow.

Q.41: Why did you override service() method in BaseCtl class?
Ans. To perform Generic worl flow in application.

Q.42: Why do you have two methods doGet and doPost in each controller ?
Ans. For view logic we use doGet & for submit logic we use doPost.

Q.43: What is the naming convention of controller?
Ans. Same as java class + suffix in Ctl. E.g. : BaseCtl.java

Q.44: Why do you always go to controller before going to view?
Ans. According to MVC guidelines:
We cannot directly access a view. View is always accessed by us via controller. Controller is responsible to fetch data from model to display at view.

Q.45: How do you perform programmatic input validation in controller?
Ans. By overriding validate() of BaseCtl in each controller.

Q.46: What is the scope of input validation error messages?
Ans. request

Q.47: which cases validate() method will not be called?
Ans. 5 cases : isNotNull(), View, Cancel, Reset, Delete

Q.48: What is pre-load data?
Ans. Data loaded at the time of HTML page loading.

Q.49: How do you load pre-loaded data?
Ans. By overriding preload() of BaseCtl.
*See code also.

Q.50: How do you convert request parameters into bean?
Ans. By populateBean() of BaseCtl.

Q.51: How do you apply Front Controller to entire application?
Ans. By applying wild card mapping to url.

Q.52: What is the responsibility of Front Controller?
Ans. It prevents any user to access application without login.

Q.53: Why have you defined urls of all views and controllers in an interface?

Ans. Because in an Interface, attributes are by default static and common and provides the loose coupling between views and controllers.

Q.54: How do you apply pagination in app?

Ans. We get page size (=10) from system.proerties and page number from view by the formula

pageNo = (pageNo – 1) * pageSize.

And append "limit pageNo, pageSize" in SQL query in model methods.

Q.55: How do you send email notification from controller?

Ans. By sendMail() of EmailUtility class.

#Views

Q.56: What type of logic are contained by a view?

Ans. View Logic (Display Logic)

Q.57: How do you create html list in the view?

Ans. By HashMap object and getList() of HTMLUtility. E.g.

```
<th>Gender<font color="red">*</font></th>
                                        <td>
                                                <%
                                                        HashMap map = new HashMap();
                                                        map.put("", "----Select----");
                                                        map.put("Male", "Male");
                                                        map.put("Female", "Female");

                                                        String htmlList = HTMLUtility.getList("gender", bean.getGender(),
map);
                                                %> <%=htmlList%>  <font
color="red"><%=ServletUtility.getErrorMessage("gender", request)%></font>
                                        </td>
```

Q.58: How do you display error messages in view?

Ans. By getErrorMessage() of ServletUtility class. E.g. :

```
<font color="red"><%=ServletUtility.getErrorMessage("gender", request)%></font>
```

Q.59: Why do you use HTMLUtility class in a view?

Ans. HTML Utility class to produce HTML contents like Dropdown List.

Q.60: How do you apply search on a view?

Ans. By input field and a submit button. E.g.

```
<td align="center"><label> First Name :</label>
<input type="text" name="firstName"
value="<%=ServletUtility.getParameter("firstName", request)%>">;
```

Q.61: How do you format and print a date at view?

Ans. By use date picker function. E.g.

```
<th align="left">Date Of Birth<font color="red">*</font></th>
                                        <td><input type="text" name="dob" class="form-control"
                                                id="datepicker"
                                                value="<%=DataUtility.getDateString(bean.getDob())%>"
                                                placeholder="yyyy/MM/dd"><a
                                                href="javascript:getCalendar(document.forms[0].dob);"> </a> <font
                                                color="red"><%=ServletUtility.getErrorMessage("dob",
request)%></font></td>
                                    </tr>
```

Q.62: What changes will you make on view, controller and model if you have to add one more search field at view?

Ans. View: make a text field and a submit button.

Controller: Populate the field by overriding populateBean() of BaseCtl class.

Model: Append that value through SQL injection in SQL query of respective method of that model.

#Models

Q.63: What type of logic are contained by a model?

Ans. Integration Logic, DataAccess Loagic, Bussiness Logic.

Q.64: Why do you use BaseModel class?

Ans. we didn't use it.

Q.65: How do you handle transaction in model?

Ans. Connection Object conn.setAutoCommit(false), conn.rollback(), conn.commit()

Q.66: What query will be generated in case of pagination?

Ans. SELECT * FROM faculty WHERE 1=1 Limit 1,10;

#Container

Q.67: Which server are you using in your application?

Ans. JBoss

Q.68: How do you get object of session in your controller?

Ans. By craeating HttpSession Object. E.g. :

HttpSession session = request.getSession(true);

Q.69: How do you track session?

Ans. 3 ways: Cookies, Hidden Form Field, URLrewriting

In our application browser itself track session by cookies.

Q.70: How do you configure Front Controller?
Ans. @WebFilter(filterName = "FrontCtl", urlPatterns = { "/ctl/*", "/doc/*" })

Q.71: How do you make a build and what is the extension of build file?
Ans.

Q.72: Where do you deploy your build?
Ans.

#Tools

Q.73: How do you do code versioning?
Ans.

Q.74: How do you checkout code from Git?
Ans.

Q.75: How do you check-in code to Git?
Ans.

Q.76: What is branching in Git?
Ans.

#Code

Q.77: Must be able to explain logic of code and classes of implemented Usecase.
Q.78: Must be able to make build war file and deploy before faculty.