

NCP Project Final Review
GROUP NUMBER: 4

S.No	Name	Roll Number
1	Ajay Pranav	CB.EN.U4CSE17606
2	Mathana Kumar S	CB.EN.U4CSE17635
3	Mithila M	CB.EN.U4CSE17638
4	Janardhana Reddy	CB.EN.U4CSE17639
5	Muthukumaaran SK	CB.EN.U4CSE17641
6	Sai Sanjay S	CB.EN.U4CSE17654

Project title : **Higher Studies/Competitive Exams**

ABSTRACT

This project aims to develop an application that helps to ease the process of information attainment regarding higher education. The importance of higher education is something that is very apparent. It paves a way to a prosperous career and innately improves the quality of life. A proper higher education shapes one's career and there are many factors regarding this. Deciding the ideal university will normally involve a lot of hassle owing to multiple factors. This web application helps to reduce the hassle by reducing the effort required to gain information regarding it by including an amalgam of specifications in a single webpage.

There are multiple, sequential modules. A login-register feature for the user(student in this case) to get a more personalized blend of information. Evidently, the user has to log in every time to access the application. The user can then go on to select a stream of their interest(Science, Finance, etc.) and the country of their choice. Based on the respective choices, the user has the option to select examinations or universities based on their choices. They can then select an examination/university and gain basic information about it. There are additional features that involve a rank/college predictor which returns the possible university based on the marks obtained. A reminder feature which reminds the user regarding upcoming opportunities based on their interests and much more. The basic static information display along with dynamic support establishes an easy and efficient way to make decisions regarding the career-defining component that is Higher Education.

WEB DESIGNING COMPONENTS

Front-end technologies involved:

1. HTML
2. CSS
 - Framework: Bulma CSS
3. Vanilla JavaScript

Back-end technologies involved:

1. JSP
2. MySQL DB

HTML: HTML (HyperText Markup Language) is the most basic building block of the Web. It defines the meaning and structure of web content. They can include writing, links, pictures, and even sound and video. HTML is used to mark and describe each of these kinds of content so the web browser can display them correctly.

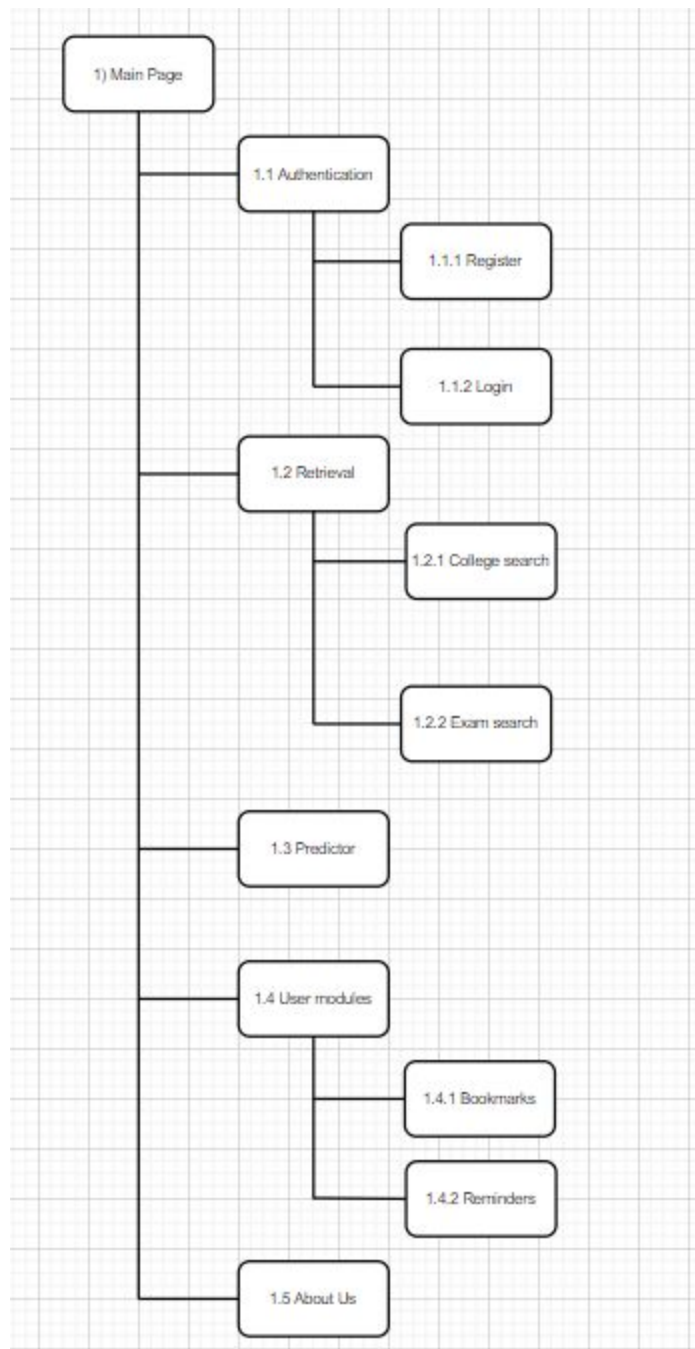
CSS: Cascading Style Sheets (CSS) is a simple mechanism for adding style (e.g., fonts, colors, spacing) to Web documents. CSS is designed to enable the separation of presentation and content, including layout, colour and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics.

JS: JavaScript (JS) is a full-fledged dynamic programming language that can add interactivity to a website. JavaScript is also commonly used in server-side programming through platforms like Node.js, or "embedded" in non-JavaScript applications where the base programming language lacks the high-level functionality that JavaScript offers.

JSP: Java Server Pages (JSP) is a server-side programming technology that enables the creation of dynamic, platform-independent methods for building

Web-based applications. JSP has access to the entire family of Java APIs, including the JDBC API to access enterprise databases.

DESIGN DIAGRAM



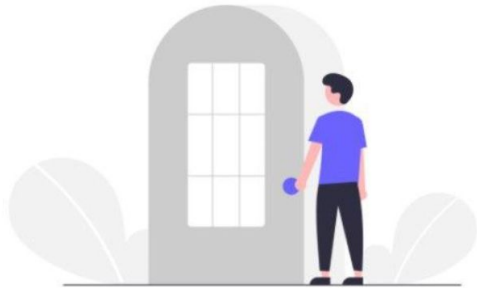
IMPLEMENTATION

Screenshots

1. Register Screen

- Serves as a user data collection point for receiving reminders.

Exam E



Register Here

Name

Email

Password

Confirm Password

Register

2. Login Screen

- Credentials checked and user entry allowed.



Welcome Back

Email

capturesmk@gmail.com

Password

secret

New User? Register Here

Login

3. Dashboard Screen

- Contains 4 main components of the WEB APP: Bookmarks, Searching courses, exams and rank predictor.

|

Logout

My Dashboard



My Bookmarks

Find your Saved Bookmarks here



Courses

Search exams under your desired course



Exams

Search your desired exams in one go.

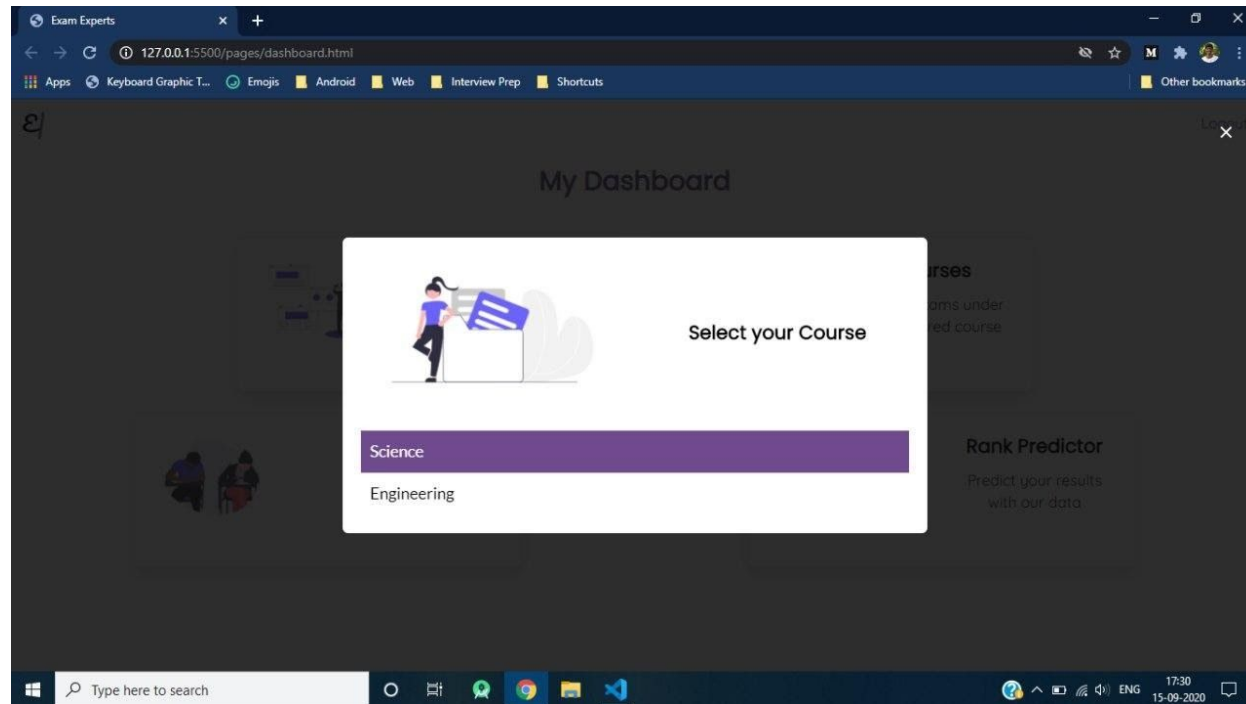


Rank Predictor

Predict your results with our data

4. Stream Selection

- Enables the user to select a stream of his/her choice which serves as a foundation in gaining knowledge based on the user's interest.




5. College Enumeration

- Displays a list of colleges that harmonize with the constraints selected by the user from which the user can get to know more about the college of his/her interest.

Exam Experts
Logout

Filters
Masters of Technology
United States Of America
Apply Filter



Northern University


Northeastern University (NU or NEU) is a private research university in Boston, Massachusetts, established in 1898. The university offers undergraduate and graduate programs on its main campus in Boston.

Examination GRE

Cut-Off Score 299

Locate in Maps View More

Save to Bookmarks



Indian Institute of Technology

Indian Institute of Technology is one of the top universities created to be Centres of Excellence for training, research and development in science, engineering and technology in India.

Examination GATE

Cut-Off Score 77

Locate in Maps View More

Save to Bookmarks

Code Snippets: Dashboard Segment

```
<body>
  <nav class="navbar is-transparent is-fixed-top" role="navigation"
    aria-label="main navigation" id="navbar">
    <div class="navbar-menu is-active">
      <div class="navbar-start">
        <!-- navbar items -->
        <div class="navbar-item" id="navtext"></div>
      </div>
      <div class="navbar-end">
        <div class="navbar-item">
          <a href="" id="logout">Logout</a>
        </div>
      </div>
    </div>
  </nav>
  <section class="section">
    <div class="mainhead has-text-centered">
      My Dashboard
    </div>
    <div class="columns">
      <div class="column is-4 is-offset-2">
        <div class="box">
          <div class="columns">
            <div class="column is-6">
              <figure class="image is-128x128">
                
              </figure>
            </div>
            <div class="column has-text-centered col-head">
              My Bookmarks
              <div class="col-text">
                Find your Saved Bookmarks here
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
```



```

        </div>
    </div>
</div>
<div class="column is-4">
    <div class="box">
        <div class="columns">
            <div class="column is-6">
                <figure class="image is-128x128">
                    
                </figure>
            </div>
            <div class="column has-text-centered col-head">
                Courses
                <div class="col-text">
                    Search exams under your desired course
                </div>
            </div>
        </div>
    </div>
</div>
</div>
</div>
<div class="columns">
    <div class="column is-4 is-offset-1">
        <div class="box">
            <div class="columns">
                <div class="column is-6">
                    <figure class="image is-128x128">
                        
                    </figure>
                </div>
                <div class="column has-text-centered col-head">
                    Exams
                    <div class="col-text">
                        Search your desired exams in one go.
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>

```

```

        </div>
    </div>

    </div>
</div>
<div class="column is-4 is-offset-2">
    <div class="box">
        <div class="columns">
            <div class="column is-6">
                <figure class="image is-128x128">
                    
                </figure>
            </div>
            <div class="column has-text-centered col-head">
                Rank Predictor
                <div class="col-text">
                    Predict your results with our data
                </div>
            </div>
        </div>
    </div>
</div>
</div>
</div>
</div>
</section>
<!--Scripts ARENA-->
<script
src="https://unpkg.com/typewriter-effect@latest/dist/core.js"></script>
<script>
    var app = document.getElementById('navtext');
    var typewriter = new Typewriter(app, {
        loop: true
    });
    typewriter.typeString('Exam Experts')
        .start();

```

```
</script>  
<!--Scripts ARENA Ends-->  
</body>
```

TECHNOLOGIES ADOPTED

XML: Extensible Markup Language (XML) is a simple, very flexible text format derived from SGML (ISO 8879). Originally designed to meet the challenges of large-scale electronic publishing, XML is also playing an increasingly important role in the exchange of a wide variety of data on the Web and elsewhere. XML was designed to be both human and machine readable. XML is widely used in a Service-oriented architecture (SOA). Disparate systems communicate with each other by exchanging XML messages. The message exchange format is standardised as an XML schema (XSD). This is also referred to as the canonical schema. XML has come into common use for the interchange of data over the Internet. [IETF RFC:3023](#), now superseded by [RFC:7303](#), gave rules for the construction of Internet Media Types for use when sending XML. It also defines the media types `application/xml` and `text/xml`, which say only that the data is in XML, and nothing about its semantics. XML documents consist entirely of characters from the Unicode repertoire. Except for a small number of specifically excluded control characters, any character defined by Unicode may appear within the content of an XML document. XML includes facilities for identifying

the *encoding* of the Unicode characters that make up the document, and for expressing characters that, for one reason or another, cannot be used directly.

The XML specification defines an XML document as a well-formed text, meaning that it satisfies a list of syntax rules provided in the specification. Some key points in the fairly lengthy list include:

- The document contains only properly encoded legal Unicode characters.
- None of the special syntax characters such as `<` and `&` appear except when performing their markup-delineation roles.
- The start-tag, end-tag, and empty-element tag that delimit elements are correctly nested, with none missing and non overlapping.
- Tag names are case-sensitive; the start-tag and end-tag must match exactly.
- Tag names cannot contain any of the characters `!"#$%&'()*+,-./:;<=>?@[\\]^_{|}~`, nor a space character, and cannot begin with `"-`, `"."`, or a numeric digit.
- A single root element contains all the other elements.

The XML specification defines a valid XML document as a well-formed XML document which also conforms to the rules of a Document Type Definition (DTD).

In addition to being well-formed, an XML document may be *valid*. This means that it contains a reference to a Document Type Definition (DTD), and that its elements and attributes are declared in that DTD and follow the grammatical rules for them that the DTD specifies.

XML processors are classified as *validating* or *non-validating* depending on whether or not they check XML documents for validity. A processor that discovers a validity error must be able to report it, but may continue normal processing.

A DTD is an example of a *schema* or *grammar*. Since the initial publication of XML 1.0, there has been substantial work in the area of schema languages for XML. Such schema languages typically constrain the set of elements that may

be used in a document, which attributes may be applied to them, the order in which they may appear, and the allowable parent/child relationships.

A newer schema language, described by the W3C as the successor of DTDs, is XML Schema, often referred to by the initialism for XML Schema instances, XSD (XML Schema Definition). XSDs are far more powerful than DTDs in describing XML languages. They use a rich data typing system and allow for more detailed constraints on an XML document's logical structure. XSDs also use an XML-based format, which makes it possible to use ordinary XML tools to help process them.

xs:schema element that defines a schema:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>  
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"></xs:schema>
```

JDBC: JDBC stands for Java Database Connectivity. JDBC is a Java API to connect and execute the query with the database. It is a part of JavaSE (Java Standard Edition). JDBC API uses JDBC drivers to connect with the database. We can use JDBC API to access tabular data stored in any relational database. By the help of JDBC API, we can save, update, delete and fetch data from the database. It is like Open Database Connectivity (ODBC). Before JDBC, ODBC API was the database API to connect and execute the query with the database. But, ODBC API uses ODBC driver which is written in C language (i.e. platform dependent and unsecured). That is why Java has defined its own API (JDBC API) that uses JDBC drivers (written in Java language).

We can use JDBC API to handle database using Java program and can perform the following activities:

1. Connect to the database
2. Execute queries and update statements to the database
3. Retrieve the result received from the database

Today, there are five types of JDBC drivers in use:

- Type 1: JDBC-ODBC bridge
- Type 2: partial Java driver
- Type 3: pure Java driver for database middleware
- Type 4: pure Java driver for direct-to-database
- Type 5: highly-functional drivers with superior performance

JDBC ('Java Database Connectivity') allows multiple implementations to exist and be used by the same application. The API provides a mechanism for dynamically loading the correct Java packages and registering them with the JDBC Driver Manager. The Driver Manager is used as a connection factory for creating JDBC connections. JDBC connections support creating and executing statements. These may be update statements such as SQL's CREATE, INSERT, UPDATE and DELETE, or they may be query statements such as SELECT. Additionally, stored procedures may be invoked through a JDBC connection. Update statements such as INSERT, UPDATE and DELETE return an update count that indicates how many **rows** were affected in the database. These statements do not return any other information. Query statements return a JDBC row result set. The row result set is used to walk over the result set. Individual columns in a row are retrieved either by name or by column number. There may be any number of rows in the result set. The row result set has metadata that describes the names of the columns and their types. JDBC connections are often managed via a connection pool rather than obtained directly from the driver. If a database operation fails, JDBC raises an `SQLException`. There is typically very little one can do to recover from such an

error, apart from logging it with as much detail as possible. It is recommended that the `SQLException` be translated into an application domain exception (an unchecked one) that eventually results in a transaction rollback and a notification to the user.

DATABASE DESIGN (ER DIAGRAM)

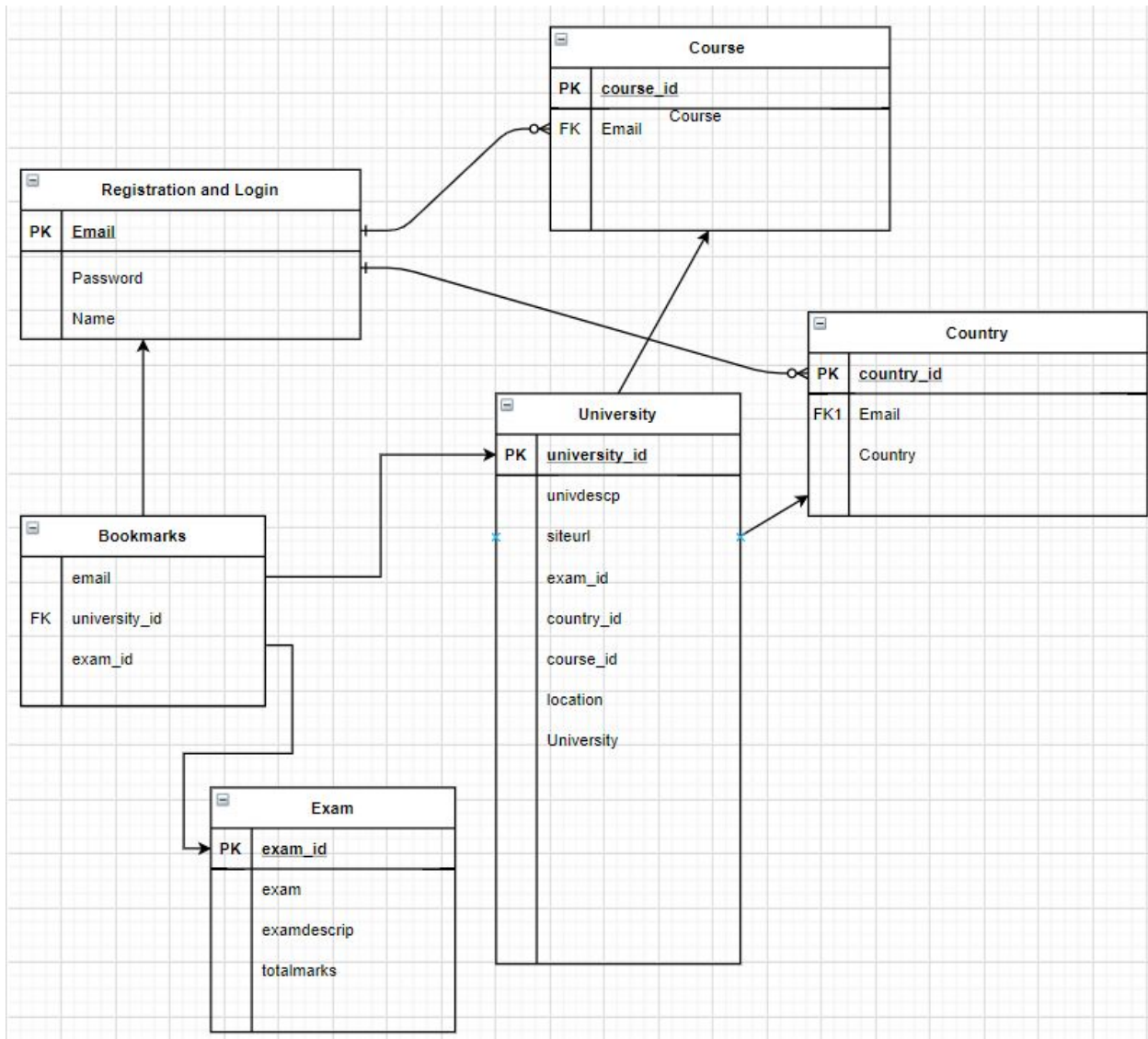


TABLE SPECIFICATIONS

Userdetails.xml

S.No	Field	Type	Purpose
1	Email	String	To enable personalised user experience by being an authorization metric
2	Password	String	Authorization metric
3	Name	String	Personalised user experience

Countrydetails.xml

S.No	Field	Type	Purpose
1	Country ID	String	To uniquely identify countries
2	Country Name	String	To understand which country the university is in

Coursedetails.xml

S.No	Field	Type	Purpose
------	-------	------	---------

1	Course ID	String	To uniquely identify the courses
2	Course Name	String	To understand the course desired by the user.

Exambookmarks.xml

S.No	Field	Type	Purpose
1	Email	String	To store bookmarks correctly for the appropriate user.
2	Exam ID	String	To store the exam for which user requires a bookmark.

Examdetails.xml

S.No	Field	Type	Purpose
------	-------	------	---------

1	Exam ID	String	To uniquely identify the exams.
2	Exam Name	String	To understand the exam desired by the user.
3	Description	String	To gain knowledge about the respective exam.
4	Total Marks	Integer	To gain knowledge about the maximum marks scorable.

Univbookmarks.xml

S.No	Field	Type	Purpose
1	Email	String	To store bookmarks correctly for the appropriate user.
2	University ID	String	To store the university for which user requires a bookmark.

Universitydetails.xml

S.No	Field	Type	Purpose
------	-------	------	---------

1	University ID	String	To uniquely identify the universities
2	University Name	String	To understand the university desired by the user.
3	Description	String	To gain knowledge about the respective university.
4	URL	String	A link to the official website of the university.
5	Location	String	A link which directs the user to a Google Maps representation of the location of the university.
6	Cutoff	Integer	To understand the threshold mark to be scored to be eligible.
7	Exam ID	String	To uniquely identify the exam applicable for the university.
8	Country ID	String	To uniquely identify the country in which the university is present.
9	Course ID	String	To uniquely identify the course offered by the university.

STEPS IN DATABASE CONNECTIVITY

There are 5 steps to connect any java application with the database using JDBC. These steps are as follows:

- Register the Driver class
- Create connection
- Create statement
- Execute queries
- Close connection

Register the Driver class: Here, Java program is loading oracle driver to establish database connection. The syntax is as follows :-

```
Class.forName("com.mysql.cj.jdbc.Driver");
```

Create Connection: The getConnection() method of DriverManager class is used to establish connection with the database. The syntax is as follows:-

```
Connection c=DriverManager.getConnection(  
    "jdbc:mysql://localhost:3306/ncp","root","password");
```

Create Statement: The createStatement() method of Connection interface is used to create the statement. The object of statement is responsible to execute queries with the database. The syntax is as follows:-

Statement s=c.createStatement();

Execute the query: The executeQuery() method of Statement interface is used to execute queries to the database. This method returns the object of ResultSet that can be used to get all the records of a table. The syntax is as follows:-

ResultSet rs=s.executeQuery("select * from Exam");

while(rs.next()){

Element student = doc.createElement("Exam");

Element examid = doc.createElement("ExamID");

examid.setTextContent(rs.getString("examID"));

student.appendChild(examid);

Element exam = doc.createElement("Exam");

exam.setTextContent(rs.getString("exam"));

student.appendChild(exam);

rootElement.appendChild(student);

}

Close connection: By closing the connection object statement and ResultSet will be closed automatically. The close() method of Connection interface is used to close the connection. The syntax is as follows:- **c.close();**

SCREENSHOTS

Bookmarks

Exa/

Logout

Bookmarked Colleges

Bookmarked Exams

Graduate Record Examination

Delete from Bookmarks

The Graduate Record Examinations is a standardized test that is an admissions requirement for many graduate schools in the United States and Canada.

Total Marks 340

Colleges

Exam Experts/

Logout

Filters Masters of Technology United States Of America Apply Filter



Northern University

Save to Bookmarks

Northeastern University (NU or NEU) is a private research university in Boston, Massachusetts, established in 1898. The university offers undergraduate and graduate programs on its main campus in Boston.

Examination GRE

Cut-Off Score 299

Locate in Maps View More



Indian Institute of Technology

Save to Bookmarks

Indian Institute of Technology is one of the top universities created to be Centres of Excellence for training, research and development in science, engineering and technology in India.

Examination GATE

Cut-Off Score 77

Locate in Maps View More

Exams

Graduate Record Examination

[Save to Bookmarks](#)

The Graduate Record Examinations is a standardized test that is an admissions requirement for many graduate schools in the United States and Canada.

Total Marks 340

Graduate Aptitude Test in Engineering

[Save to Bookmarks](#)

The Graduate Aptitude Test in Engineering (GATE) is an examination that primarily tests the comprehensive understanding of various undergraduate subjects in engineering

Total Marks 100

International English Language Testing System

[Save to Bookmarks](#)

The International English Language Testing System, or IELTS, is an international standardized test of English language proficiency for non-native English language speakers

Total Marks 9

Test of English as a Foreign Language

[Save to Bookmarks](#)

XML Schema Specification

countrydetails.xsd

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="Countries">
    <xs:complexType>
      <xs:sequence>
        <xs:element maxOccurs="unbounded" name="Country">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="CountryID" type="xs:string" />
              <xs:element name="CountryName" type="xs:string" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

coursedetails.xsd

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="Courses">
    <xs:complexType>
      <xs:sequence>
        <xs:element maxOccurs="unbounded" name="Course">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="CourseID" type="xs:string" />
              <xs:element name="CourseName" type="xs:string" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

```

        </xs:element>
    </xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>

```

exambookmarks.xsd

```

<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:element name="Bookmarks">
        <xs:complexType>
            <xs:sequence>
                <xs:element maxOccurs="unbounded" name="Bookmark">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name="Email" type="xs:string" />
                            <xs:element name="ExamID" type="xs:string" />
                            <xs:element name="ExamName" type="xs:string" />
                            <xs:element name="Description" type="xs:string" />
                            <xs:element name="Totalmarks" type="xs:unsignedShort" />
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
</xs:schema>

```

examdetails.xsd

```

<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:element name="Exams">
        <xs:complexType>
            <xs:sequence>

```

```

    <xs:element maxOccurs="unbounded" name="Exam">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="ExamID" type="xs:string" />
          <xs:element name="ExamName" type="xs:string" />
          <xs:element name="Description" type="xs:string" />
          <xs:element name="Totalmarks" type="xs:unsignedShort" />
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>

```

univbookmarks.xsd

```

<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="Bookmarks">
    <xs:complexType>
      <xs:sequence>
        <xs:element maxOccurs="unbounded" name="Bookmark">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Email" type="xs:string" />
              <xs:element name="UniversityID" type="xs:string" />
              <xs:element name="University" type="xs:string" />
              <xs:element name="Description" type="xs:string" />
              <xs:element name="URL" type="xs:string" />
              <xs:element name="Location" type="xs:string" />
              <xs:element name="Cutoff" type="xs:unsignedByte" />
              <xs:element name="ExamID" type="xs:string" />
              <xs:element name="CountryID" type="xs:string" />
              <xs:element name="CourseID" type="xs:string" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>

```

```

        </xs:element>
    </xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>

```

universitydetails.xsd

```

<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:element name="Universities">
        <xs:complexType>
            <xs:sequence>
                <xs:element maxOccurs="unbounded" name="University">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name="UniversityID" type="xs:string" />
                            <xs:element name="UniversityName" type="xs:string" />
                            <xs:element name="Description" type="xs:string" />
                            <xs:element name="URL" type="xs:string" />
                            <xs:element name="Location" type="xs:string" />
                            <xs:element name="Cutoff" type="xs:unsignedShort" />
                            <xs:element name="ExamID" type="xs:string" />
                            <xs:element name="CountryID" type="xs:string" />
                            <xs:element name="CourseID" type="xs:string" />
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
</xs:schema>

```

userdetails.xsd

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="users">
    <xs:complexType>
      <xs:sequence>
        <xs:element maxOccurs="unbounded" name="user">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="email" type="xs:string" />
              <xs:element name="pass" type="xs:string" />
              <xs:element name="name" type="xs:string" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Displaying XML with CSS features

countrydetails.xsl

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
<html>
<body>
  <h2>Countries</h2>
  <table border="1">
    <tr bgcolor="#9acd32">
      <th style="text-align:left">Country ID</th>
      <th style="text-align:left">CountryName</th>
    </tr>
    <xsl:for-each select="countries/country">
      <tr>
        <td><xsl:value-of select="countryid"/></td>
        <td><xsl:value-of select="countryname"/></td>
      </tr>
    </xsl:for-each>
  </table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```

coursedetails.xsl

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
<html>
<body>
```

```

<h2>Courses</h2>
<table border="1">
  <tr bgcolor="#9acd32">
    <th style="text-align:left">Course ID</th>
    <th style="text-align:left">Course Name</th>
  </tr>
  <xsl:for-each select="Courses/Course">
    <tr>
      <td><xsl:value-of select="CourseID"/></td>
      <td><xsl:value-of select="CourseName"/></td>
    </tr>
  </xsl:for-each>
</table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>

```

exambookmarks.xsl

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
<html>
<body>
  <h2>Bookmarks</h2>
  <table border="1">
    <tr bgcolor="#9acd32">
      <th style="text-align:left">Email</th>
      <th style="text-align:left">ExamID</th>
      <th style="text-align:left">ExamName</th>
      <th style="text-align:left">Description</th>
      <th style="text-align:left">Totalmarks</th>
    </tr>
    <xsl:for-each select="Bookmarks/Bookmark">
      <tr>

```

```

        <td><xsl:value-of select="Email"/></td>
        <td><xsl:value-of select="ExamID"/></td>
        <td><xsl:value-of select="ExamName"/></td>
        <td><xsl:value-of select="Description"/></td>
        <td><xsl:value-of select="Totalmarks"/></td>
    </tr>
</xsl:for-each>
</table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>

```

examdetails.xsl

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
<html>
<body>
    <h2>Exams</h2>
    <table border="1">
        <tr bgcolor="#9acd32">
            <th style="text-align:left">ExamID</th>
            <th style="text-align:left">ExamName</th>
            <th style="text-align:left">Description</th>
            <th style="text-align:left">Totalmarks</th>
        </tr>
        <xsl:for-each select="Exams/Exam">
            <tr>
                <td><xsl:value-of select="ExamID"/></td>
                <td><xsl:value-of select="ExamName"/></td>
                <td><xsl:value-of select="Description"/></td>
                <td><xsl:value-of select="Totalmarks"/></td>
            </tr>
        </xsl:for-each>
    </table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>

```



```

    </table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>

```

univbookmarks.xsl

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
<html>
<body>
    <h2>Bookmarks</h2>
    <table border="1">
        <tr bgcolor="#9acd32">
            <th style="text-align:left">Email</th>
            <th style="text-align:left">UniversityID</th>
            <th style="text-align:left">University</th>
            <th style="text-align:left">Description</th>
            <th style="text-align:left">URL</th>
            <th style="text-align:left">Location</th>
            <th style="text-align:left">Cutoff</th>
            <th style="text-align:left">ExamID</th>
            <th style="text-align:left">CountryID</th>
            <th style="text-align:left">CourseID</th>
        </tr>
        <xsl:for-each select="Bookmarks/Bookmark">
            <tr>
                <td><xsl:value-of select="Email"/></td>
                <td><xsl:value-of select="UniversityID"/></td>
                <td><xsl:value-of select="University"/></td>
                <td><xsl:value-of select="Description"/></td>
                <td><xsl:value-of select="URL"/></td>
                <td><xsl:value-of select="Location"/></td>
                <td><xsl:value-of select="Cutoff"/></td>
            
```

```

        <td><xsl:value-of select="ExamID"/></td>
        <td><xsl:value-of select="CountryID"/></td>
        <td><xsl:value-of select="CourseID"/></td>
    </tr>
</xsl:for-each>
</table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>

```

universitydetails.xsl

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
<html>
<body>
    <h2>Universities</h2>
    <table border="1">
        <tr bgcolor="#9acd32">
            <th style="text-align:left">UniversityID</th>
            <th style="text-align:left">University</th>
            <th style="text-align:left">Description</th>
            <th style="text-align:left">URL</th>
            <th style="text-align:left">Location</th>
            <th style="text-align:left">Cutoff</th>
            <th style="text-align:left">ExamID</th>
            <th style="text-align:left">CountryID</th>
            <th style="text-align:left">CourseID</th>
        </tr>
        <xsl:for-each select="Universities/University">
        <tr>
            <td><xsl:value-of select="UniversityID"/></td>
            <td><xsl:value-of select="University"/></td>
            <td><xsl:value-of select="Description"/></td>

```

```

        <td><xsl:value-of select="URL"/></td>
        <td><xsl:value-of select="Location"/></td>
        <td><xsl:value-of select="CountryID"/></td>
        <td><xsl:value-of select="CourseID"/></td>
    </tr>
</xsl:for-each>
</table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>

```

userdetails.xsl

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
<html>
<body>
    <h2>Countries</h2>
    <table border="1">
        <tr bgcolor="#9acd32">
            <th style="text-align:left">email</th>
            <th style="text-align:left">pass</th>
            <th style="text-align:left">name</th>
        </tr>
        <xsl:for-each select="users/user">
            <tr>
                <td><xsl:value-of select="email"/></td>
                <td><xsl:value-of select="pass"/></td>
                <td><xsl:value-of select="name"/></td>
            </tr>
        </xsl:for-each>
    </table>
</body>
</html>

```

```
</xsl:template>  
</xsl:stylesheet>
```

Integrating XML,HTML and Backend tables

bookmarks.html

```
<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Exam Experts</title>
    <link rel="stylesheet" href="../css/bulma.min.css">
    <link rel="stylesheet" href="../css/main.css">
    <link
href="https://fonts.googleapis.com/css2?family=Lato&family=Pacifico&family=Poppins
&family=Quicksand&display=swap" rel="stylesheet">
  </head>
  <body onload="fetchXML()">
    <nav class="navbar is-transparent is-fixed-top" role="navigation"
aria-label="main navigation" id="navbar">
      <div class="navbar-menu is-active">
        <div class="navbar-start">
          <!-- navbar items -->
          <div class="navbar-item" id="navtext"></div>
        </div>
        <div class="navbar-end">
          <div class="navbar-item">
            <a href="" id="logout">Logout</a>
          </div>
        </div>
      </div>
    </nav>
    <section class="section">
      <div class="mainhead has-text-centered"><div class="box">Bookmarked
Colleges</div></div>
      <div id="clgbk">
      </div>
      <div class="mainhead has-text-centered"><div class="box">Bookmarked
Exams</div></div>
      <div id="exambk">
```

```

</div>
</section>
<script
src="https://unpkg.com/typewriter-effect@latest/dist/core.js"></script>
<script src =
"https://ajax.googleapis.com/ajax/libs/jquery/2.1.1/jquery.min.js"></script>
<script>

    var app = document.getElementById('navtext');
    var typewriter = new Typewriter(app, {
        loop: true
    });
    typewriter.typeString('Exam Experts')
        .start();

    function myExams(xml) {
        var i;
        var xmlDoc = xml.responseXML;
        var x = xmlDoc.getElementsByTagName("Bookmark");

    }

    $(document).ready(function() {
    var xml;
    var c=0;
    $.get('../xml/exambookmarks.xml', null, function (data, textStatus) {
    xml=data;
    $(xml).find('Bookmark').each( function() {
        var item = $(this);
        if(item.find('Email').text() == sessionStorage.getItem("emailID"))
        {
            var htmlbase = '<div class="columns is-vcentered"
id="exam-list"><div class="column side-align"><div class="box"><span class="tag
is-link is-medium bookmark"><a>Delete from Bookmarks</a></span><div
class="exam-name">';
            var res='';
            res += htmlbase+ item.find("ExamName").text() + "</div>" + '<p
class="exam-desc">' + item.find("Description").text() + ' <div
class="exam-details"><span class="tag is-medium is-warning desc-tags">Total
Marks</span><span class="exam-cutoff">'+
            item.find("Totalmarks").text() + '</span></div></div></div></div>';

```

```

        $("#exambk").append(res);
    }
    });
});

$.get('../xml/univbookmarks.xml', null, function (data, textStatus) {
    xml=data;
    $(xml).find('Bookmark').each( function(){
        var item = $(this);
        if(item.find('Email').text()==sessionStorage.getItem("emailID"))
        {
            var htmlbase = '<div class="columns is-vcentered'
id="college-list"><div class="column is-2"></div><div class="column side-align"><div class="box"><span class="tag
is-link is-medium bookmark"><a>Delete from Bookmarks</a></span><div
class="clg-name">'+ item.find("UniversityName").text() + '</div><p
class="clg-desc">' + item.find("Description").text() + '</p><div
class="clg-details"><div class="space-tags"><span class="tag is-medium is-warning
desc-tags">Examination</span><span class="clg-exam">' +item.find("ExamID").text()
+'</span></div><div><span class="tag is-medium is-warning desc-tags">Cut-Off
Score</span><span class="clg-cutoff">' +
item.find("Cutoff").text()+'</span></div></div><div class="buttons is-right
clg-btns"><a class="button is-dark is-rounded clg-location" target="_blank"
href="'+item.find("Location").text()+'">Locate in Maps</a><a class="button is-link
clg-url" target="_blank" href="'+item.find("URL").text()+'">View
More</a></div></div></div></div>';
            $("#clgbk").append(res);
        }
    });
});

});
</script>
</body>
</html>

```

XML Code Snippets

Countrydetails.xml

```
<?xml version="1.0" encoding="UTF-8"
standalone="no"?>
<countries>
  <country>
    <countryid>USA</countryid>
    <countryname>United States Of
America</countryname>
  </country>
  <country>
    <countryid>FRA</countryid>

<countryname>France</countryname>
>
  </country>
  <country>
    <countryid>GER</countryid>

<countryname>Germany</countryname>
me>
  </country>
  <country>
    <countryid>RUS</countryid>

<countryname>Russia</countryname>
>
  </country>
  <country>
    <countryid>IND</countryid>

<countryname>India</countryname>
  </country>
</countries>
```

Coursedetails.xml

```
<?xml version="1.0"
encoding="UTF-8"
standalone="no"?>
<Courses>
  <Course>
    <CourseID>M.Tech</CourseID>
    <CourseName>Masters of
Technology</CourseName>
  </Course>
  <Course>
    <CourseID>MS</CourseID>
    <CourseName>Masters of
Science</CourseName>
  </Course>
  <Course>
    <CourseID>MBA</CourseID>
    <CourseName>Masters of
Business
Administration</CourseName>
  </Course>
  <Course>
    <CourseID>MA</CourseID>
    <CourseName>Masters of
Arts</CourseName>
  </Course>
  <Course>
    <CourseID>M.Com</CourseID>
    <CourseName>Masters of
Commerce</CourseName>
  </Course>
```


</Courses>

Exambookmarks.xml

```
<?xml version="1.0"
encoding="UTF-8"
standalone="no"?>
```

```
<Bookmarks>
  <Bookmark>
```

```
<Email>msmuthu.sk@gmail.com</
Email>
  <ExamID>GRE</ExamID>
  <ExamName>Graduate
Record Examination</ExamName>
  <Description>The Graduate
Record Examinations is a
standardized test that is an
admissions requirement for many
graduate schools in the United
States and Canada.</Description>
```

```
<Totalmarks>340</Totalmarks>
</Bookmark>
<Bookmark>
```

```
<Email>ajaypranav06@gmail.com<
/Email>
  <ExamID>GATE</ExamID>
  <ExamName>Graduate
Aptitude Test in
Engineering</ExamName>
  <Description>The Graduate
Aptitude Test in Engineering
(GATE) is an examination that
```

primarily tests the comprehensive understanding of various undergraduate subjects in engineering</Description>

```
<Totalmarks>100</Totalmarks>
</Bookmark>
<Bookmark>
```

```
<Email>capturesmk@gmail.com</
Email>
  <ExamID>GATE</ExamID>
  <ExamName>Graduate
Aptitude Test in
Engineering</ExamName>
  <Description>The Graduate
Aptitude Test in Engineering
(GATE) is an examination that
primarily tests the comprehensive
understanding of various
undergraduate subjects in
engineering</Description>
```

```
<Totalmarks>100</Totalmarks>
</Bookmark>
</Bookmarks>
```

Examdetails.xml

```
<?xml version="1.0"
encoding="UTF-8"
standalone="no"?>
<Exams>
  <Exam>
    <ExamID>GRE</ExamID>
    <ExamName>Graduate
Record Examination</ExamName>
    <Description>The Graduate
Record Examinations is a
standardized test that is an
admissions requirement for many
graduate schools in the United
States and Canada.</Description>

    <Totalmarks>340</Totalmarks>
  </Exam>
  <Exam>
    <ExamID>GATE</ExamID>
    <ExamName>Graduate
Aptitude Test in
Engineering</ExamName>
    <Description>The Graduate
Aptitude Test in Engineering
(GATE) is an examination that
primarily tests the comprehensive
understanding of various
undergraduate subjects in
engineering</Description>

    <Totalmarks>100</Totalmarks>
  </Exam>
</Exams>
```

```
    <ExamID>IELTS</ExamID>
    <ExamName>International
English Language Testing
System</ExamName>
    <Description>The International
English Language Testing System,
or IELTS, is an international
standardized test of English
language proficiency for
non-native English language
speakers</Description>
    <Totalmarks>9</Totalmarks>
  </Exam>
  <Exam>
    <ExamID>TOEFL</ExamID>
    <ExamName>Test of English
as a Foreign
Language</ExamName>
    <Description>Test of English
as a Foreign Language is a
standardized test to measure the
English language ability of
non-native speakers wishing to
enroll in English-speaking
universities</Description>

    <Totalmarks>120</Totalmarks>
  </Exam>
</Exams>
```

Univbookmarks.xml

```
<?xml version="1.0"
encoding="UTF-8"
standalone="no"?>
```

```

<Bookmarks>
  <Bookmark>

<Email>capturesmk@gmail.com</
Email>

<UniversityID>IIT</UniversityID>
  <University>Indian Institute of
  Technology</University>
  <Description>Indian Institute
  of Technology is one of the top
  universities created to be Centres
  of Excellence for training, research
  and development in science,
  engineering and technology in
  India. </Description>

<URL>https://www.iitsystem.ac.in/
</URL>

<Location>https://goo.gl/maps/1z
2PWDgqDuKosfez5</Location>
  <Cutoff>77</Cutoff>
  <ExamID>GATE</ExamID>
  <CountryID>IND</CountryID>
  <CourseID>M.Tech</CourseID>
</Bookmark>
<Bookmark>

<Email>sai.sanjay7@gmail.com</E
mail>

<UniversityID>LMU</UniversityID>
  <University>London
  Metropolitan
  University</University>

```

```

  <Description>London
  Metropolitan University, commonly
  known as London Met, is a public
  research university in London,
  England. The University of North
  London and London Guildhall
  University merged in 2002 to
  create the university</Description>

<URL>https://www.londonmet.ac.u
k/</URL>

<Location>https://goo.gl/maps/is
mJFceFTD1WZX3L9</Location>
  <Cutoff>6</Cutoff>
  <ExamID>IELTS</ExamID>
  <CountryID>UK</CountryID>
  <CourseID>MBA</CourseID>
</Bookmark>
</Bookmarks>

```

Universitydetails.xml

```

<?xml version="1.0"
encoding="UTF-8"
standalone="no"?>
<Universities>
  <University>

<UniversityID>NU</UniversityID>
  <UniversityName>Northern
  University</UniversityName>
  <Description>Northeastern
  University (NU or NEU) is a private
  research university in Boston,

```

Massachusetts, established in 1898. The university offers undergraduate and graduate programs on its main campus in Boston.</Description>

<URL><https://www.northeastern.edu/></URL>

<Location><https://goo.gl/maps/cprhmsut4NAZGfuZ7></Location>
<Cutoff>299</Cutoff>
<ExamID>GRE</ExamID>
<CountryID>USA</CountryID>
<CourseID>MS</CourseID>
</University>
<University>

<UniversityID>IIT</UniversityID>
<UniversityName>Indian Institute of Technology</UniversityName>
<Description>Indian Institute of Technology is one of the top universities created to be Centres of Excellence for training, research and development in science, engineering and technology in India. </Description>

<URL><https://www.iitsystem.ac.in/></URL>

<Location><https://goo.gl/maps/1z2PWDgqDuKosfez5></Location>
<Cutoff>77</Cutoff>

<ExamID>GATE</ExamID>
<CountryID>IND</CountryID>
<CourseID>M.Tech</CourseID>
</University>
<University>

<UniversityID>LMU</UniversityID>
<UniversityName>London Metropolitan University</UniversityName>
<Description>London Metropolitan University, commonly known as London Met, is a public research university in London, England. The University of North London and London Guildhall University merged in 2002 to create the university</Description>

<URL><https://www.londonmet.ac.uk/></URL>

<Location><https://goo.gl/maps/is mJFceFTD1WZX3L9></Location>
<Cutoff>6</Cutoff>
<ExamID>IELTS</ExamID>
<CountryID>UK</CountryID>
<CourseID>MBA</CourseID>
</University>
<University>

<UniversityID>ESC</UniversityID>
<UniversityName>ESC Clermont Business School</UniversityName>

<Description>ESC Clermont Business School is a business school located in France, in the city of Clermont-Ferrand. Established in 1919, the school of management is a Grande Ecole that is recognized by The French Ministry of Higher Education and Research.</Description>

<URL>https://www.esc-clermont.fr/en/</URL>

<Location>https://g.page/esc-clermont-bs?share</Location>

<Cutoff>85</Cutoff>
<ExamID>TOEFL</ExamID>
<CountryID>FRA</CountryID>
<CourseID>M.Com</CourseID>
</University>
<University>

<UniversityID>SRH</UniversityID>
<UniversityName>SRH
HOCHSCHULE
HEIDELBERG</UniversityName>
<Description>The SRH University of Applied Sciences Heidelberg is a state-recognized private university in Heidelberg. The university opened on October 1, 1969 as a "facility for rehabilitation of people with disabilities in the tertiary education sector".</Description>

<URL>https://www.hs-fresenius.de/</URL>

<Location>https://goo.gl/maps/VQCh2jpBGpGs1BT8</Location>

<Cutoff>80</Cutoff>
<ExamID>TOEFL</ExamID>
<CountryID>GER</CountryID>
<CourseID>MA</CourseID>

</University>
</Universities>

Userdetails.xml

<?xml version="1.0"
encoding="UTF-8"
standalone="no"?>
<users>
<user>

<email>sai.sanjay7@gmail.com</email>
<pass>Saisan27</pass>
<name>Sai Sanjay</name>
</user>
<user>

<email>capturesmk@gmail.com</email>
<pass>Mathan08</pass>
<name>Mathana
Kumar</name>
</user>
<user>

<email>msmuthu.sk@gmail.com</email>

<pass>Muthusk41</pass>

<name>Muthukumaaran</name>

</user>

<user>

<email>ajaypranav06@gmail.com</email>

<pass>Ajaypran06</pass>

<name>Ajay Pranav</name>

</user>

<user>

<email>mithilamarapareddy3115@gmail.com</email>

<pass>Mithila3115</pass>

<name>Mithila</name>

</user>

<user>

<email>janardhanareddymc46@gmail.com</email>

<pass>Janareddy46</pass>

<name>Janardhana

Reddy</name>

</user>

</users>

TECHNOLOGIES ADOPTED

Apache Tomcat: Apache Tomcat (called "Tomcat" for short) is an open-source implementation of the Java Servlet, JavaServer Pages, Java Expression Language and WebSocket technologies. Tomcat provides a "pure Java" HTTP web server environment in which Java code can run. Tomcat is developed and maintained by an open community of developers under the auspices of the Apache Software Foundation, released under the Apache License 2.0 license. Tomcat started off as a servlet reference implementation by James Duncan Davidson, a software architect at Sun Microsystems. He later helped make the project open-source and played a key role in its donation by Sun Microsystems to the Apache Software Foundation. The Apache Ant software build automation tool was developed as a side-effect of the creation of Tomcat as an open source project.

Davidson had initially hoped that the project would become open-sourced and, since many open-source projects had O'Reilly books associated with them featuring an animal on the cover, he wanted to name the project after an animal. He came up with *Tomcat* since he reasoned the animal represented something that could fend for itself. Although the tomcat was already in use for another O'Reilly title, his wish to see an animal cover eventually came true when O'Reilly published their Tomcat book with a snow leopard on the cover in 2003.

It began as the reference implementation for the very first Java-Server Pages and the Java Servlet API. However, it no longer works as the reference implementation for both of these technologies, but it is considered as the first choice among the users even after that. It is still one of the most widely used java-servers due to several capabilities such as good extensibility, proven core engine, and well-test and durable.

Here we used the term "servlet" many times, so what is java servlet; it is a kind of software that enables the webserver to handle the dynamic(java-based) content using the Http protocols. At heart, Tomcat is a servlet and JSP container. A Java servlet encapsulates code and business logic and defines how requests and responses should be handled in a Java

server. JSP is a server-side view rendering technology. As the developer, you write the servlet or JSP page, then let Tomcat handle the routing.

Tomcat also contains the Coyote engine, which is a web server. Thanks to Coyote, it's possible to extend Tomcat to include a variety of Java enterprise specs and capabilities, including the Java Persistence API (JPA). Tomcat also has an extended version, called TomEE, that includes more enterprise features.

Tomcat is the result of an open collaboration of developers and is available from the Apache Web site in both binary and source versions. Tomcat can be used as either a standalone product with its own internal Web server or together with other Web servers, including Apache, Netscape Enterprise Server, Microsoft Internet Information Server (IIS), and Microsoft Personal Web Server.

Tomcat has a variety of good documentation available, including a wide range of online tutorials that can be viewed or downloaded. This makes it a popular choice to fill the role of an application server in almost all Java web applications. Whether you are looking for startup settings, hardening and security guides, installation instructions, or server configuration notes, Tomcat has you covered.

Tomcat requires a Java Runtime Enterprise Environment that conforms to JRE 1.1 or later. Tomcat remains actively developed, keeping pace with change, and delivering a solid and reliable platform for deploying web apps. Both its continued popularity and choice as the default Java platform for many PaaS systems testify to its ongoing success.

JSP: Java Server Pages (JSP; formerly JavaServer Pages) is a collection of technologies that helps software developers create dynamically generated web pages based on HTML, XML, SOAP, or other document types. Released in 1999 by Sun Microsystems, JSP is similar to PHP and ASP, but uses the Java programming language.

Java Server Pages can be used independently or as the view component of a server-side model-view-controller design, normally with JavaBeans as the model and Java servlets (or a framework such as Apache Struts) as the controller. This is a type of Model 2 architecture.

JSP allows Java code and certain predefined actions to be interleaved with static web markup content, such as HTML. The resulting page is compiled and executed on the server to deliver a document. The compiled pages, as well as any dependent Java libraries, contain Java bytecode rather than machine code. Like any other .jar or Java program, code must be executed within a Java virtual machine (JVM) that interacts with the server's host operating system to provide an abstract, platform-neutral environment.

JSPs are usually used to deliver HTML and XML documents, but through the use of OutputStream, they can deliver other types of data as well

The Web container creates JSP implicit objects like request, response, session, application, config, page, pageContext, out and exception. JSP Engine creates these objects during the translation phase.

A JavaServer Pages compiler is a program that parses JSPs, and transforms them into executable Java Servlets. A program of this type is usually embedded into the application server and runs automatically the first time a JSP is accessed, but pages may also be precompiled for better performance, or compiled as a part of the build process to test for errors.

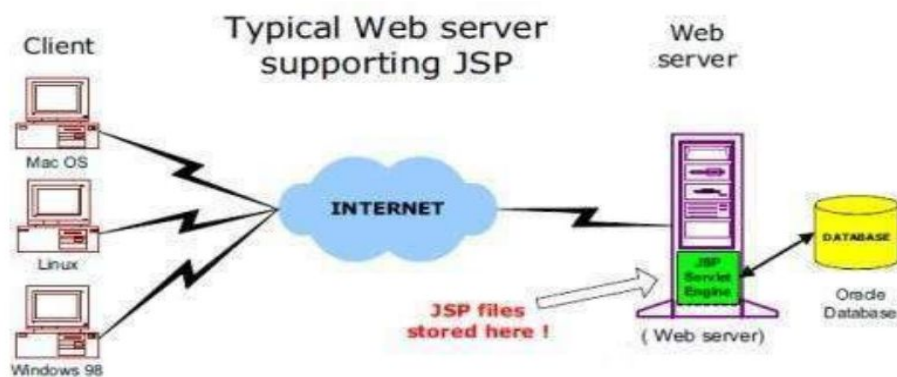
Some JSP containers support configuring how often the container checks JSP file timestamps to see whether the page has changed. Typically, this timestamp would be set to a short interval (perhaps seconds) during software development, and a longer interval (perhaps minutes, or even never) for a deployed Web application.

JavaServer Pages often serve the same purpose as programs implemented using the Common Gateway Interface (CGI). But JSP offers several advantages in comparison with the CGI.

- Performance is significantly better because JSP allows embedding Dynamic Elements in HTML Pages itself instead of having separate CGI files.
- JSP are always compiled before they are processed by the server unlike CGI/Perl which requires the server to load an interpreter and the target script each time the page is requested.
- JavaServer Pages are built on top of the Java Servlets API, so like Servlets, JSP also has access to all the powerful Enterprise Java APIs, including JDBC, JNDI, EJB, JAXP, etc.
- JSP pages can be used in combination with servlets that handle the business logic, the model supported by Java servlet template engines.

Using JSP, you can collect input from users through Web Page forms, present records from a database or another source, and create Web Pages dynamically.

JSP tags can be used for a variety of purposes, such as retrieving information from a database or registering user preferences, accessing JavaBeans components, passing control between pages, and sharing information between requests, pages etc.



JSP ARCHITECTURE

STEPS IN J2EE APPLICATION DEPLOYMENT

1. Preparation

1. Start J2EE server.

C:\> j2ee -verbose

2. Start deploytool.

C:\> deploytool

2. Creating the J2EE Application

The sample application contains two J2EE components: an enterprise bean and a Web component. Before building these components, you will create a new J2EE application called ExamExperts and will store it in an EAR file named ExamExperts.ear.

Steps:

1. In deploytool, select File->New->Application. Click Browse.
2. In the file chooser, navigate to 2tier/example/build.
3. In the File Name field, enter ExamExperts.ear.
4. Click New Application. Click OK.

3. Creating the Web Client

The Web client is contained in the JSP page, 2tier/example/src/web directory and 2tier/example/src/classes directory. A JSP page is a text-based document that contains static template data, which can be expressed in any text-based format such as HTML, WML, and XML; and JSP elements, which construct dynamic content.

Codes of the Web Client

The statements (in bold in the following code) for locating the home interface, creating an enterprise bean instance, and invoking a business method are nearly identical to those of the J2EE application client.

The classes needed by the client are declared with a JSP page directive (enclosed within the <%@ %> characters). Because locating the home interface and creating the enterprise bean are performed only once, this code appears in a JSP declaration (enclosed within the <%! %> characters) that contains the initialization method, jsplnit, of the JSP page. The

declaration is followed by standard HTML markup for creating a form with an input field. A scriptlet (enclosed within the `<% %>` characters) retrieves a parameter from the request and converts it to a double. Finally, JSP expressions (enclosed within `<%= %>` characters) invoke the enterprise bean's business methods and insert the result into the stream of data returned to the client.

Compiling the Web Client

The J2EE server automatically compiles Web clients that are JSP pages. For the Java Bean classes, you would have to compile it. Now you are ready to compile these classes, go to the `2tier/example/src/classes/` directory and type: `javac *.java`

Packaging the Web Client

To package a Web client, you run the New Web Component wizard of the `deploytool` utility. During this process the wizard performs the following tasks.

- ✓ Creates the Web application deployment descriptor
- ✓ Adds the component files to a WAR file
- ✓ Adds the WAR file to the application's `ExamExperts.ear` file.

IMPLEMENTATION

CODE SNIPPETS FOR SERVLET:

LoginServlet.java

```
package web;
import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import utils.LoginUtil;
import db.LoginDAO;

@WebServlet("/login")
public class LoginServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;
    private LoginDAO loginDao;

    public void init() {
        loginDao = new LoginDAO();
    }

    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

        String username = request.getParameter("email");
        String password = request.getParameter("pass");
        LoginUtil loginUtil = new LoginUtil();
        loginUtil.setUsername(username);
        loginUtil.setPassword(password);

        try {
            if (loginDao.validate(loginUtil)) {
                HttpSession session = request.getSession();
                session.setAttribute("email",username);
            }
        }
    }
}
```

```

        response.setStatus(200);
        response.sendRedirect("dashboard.jsp");
    } else {
        request.setAttribute("error","true");
        request.getRequestDispatcher("login.jsp").forward(request,
response);
    }
} catch (ClassNotFoundException e) {
    e.printStackTrace();
}
}
}
}

```

ExamsServlet.java

```

package web;
import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.json.simple.JSONArray;
import utils.ExamsUtil;
import db.ExamsDAO;

@WebServlet("/exams")
public class ExamsServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;
    private ExamsDAO examsDao;

    public void init() {
        examsDao = new ExamsDAO();
    }

    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

        String username = (String) request.getSession(false).getAttribute("email");
        System.out.println("EMAIL ID from Sessions Storage: "+username);

        ExamsUtil examsUtil = new ExamsUtil();
    }
}

```

```

        examsUtil.setUsername(username);

        try {
            if (examsDao.validate(examsUtil)) {
                response.setStatus(200);

                JSONArray exams = ExamsDAO.exJSON;
                request.setAttribute("exams", exams.toString());
                request.getRequestDispatcher("exams.jsp").forward(request,
response);
            } else {
                response.setStatus(401);
            }
        } catch (ClassNotFoundException e) {
            e.printStackTrace();
        }
    }

    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

        String examid = request.getParameter("eid");
        String username = (String) request.getSession(false).getAttribute("email");
        System.out.println("passed:" + examid);
        ExamsUtil examsUtil = new ExamsUtil();
        examsUtil.seteid(examid);
        examsUtil.setUsername(username);

        try {
            if (examsDao.validate(examsUtil)) {
                response.setStatus(200);
                JSONArray collegeBookmarks = examsDao.collegeJSON;
                JSONArray examBookmarks = examsDao.examJSON;

                request.setAttribute("collegeBookmarks", collegeBookmarks.toString());

                request.setAttribute("examBookmarks", examBookmarks.toString());

                request.getRequestDispatcher("bookmarks.jsp").forward(request, response);
            } else {
                request.setAttribute("error", "true");
                request.getRequestDispatcher("exams.jsp").forward(request,
response);
            }
        }
    }

```

```

        } catch (ClassNotFoundException e) {
            e.printStackTrace();
        }
    }
}

```

CollegesServlet.java

```

package web;
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import org.json.simple.JSONArray;
import utils.BookmarksUtil;
import utils.CollegesUtil;
import db.CollegesDAO;

@WebServlet("/colleges")
public class CollegesServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;
    public static CollegesDAO collegesDao;

    public void init() {
        collegesDao = new CollegesDAO();
    }

    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

        String username = (String) request.getSession(false).getAttribute("email");
        System.out.println("EMAIL ID from Sessions Storage: "+username);

        CollegesUtil collegesUtil = new CollegesUtil();
        collegesUtil.setUsername(username);

        try {
            if (collegesDao.validate(collegesUtil)) {
                response.setStatus(200);
            }
        }
    }
}

```



```

        JSONArray college = collegesDao.clgJSON;
        request.setAttribute("college",college.toString());
        request.getRequestDispatcher("colleges.jsp").forward(request,
response);
    } else {
        response.setStatus(401);
    }
} catch (ClassNotFoundException e) {
    e.printStackTrace();
}
}

protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {

    String univid = request.getParameter("uid");
    String username = (String) request.getSession(false).getAttribute("email");
    CollegesUtil collegesUtil = new CollegesUtil();
    collegesUtil.setuid(univid);
    collegesUtil.setUsername(username);

    try {
        if (collegesDao.validate(collegesUtil)) {
            response.setStatus(200);
            JSONArray collegeBookmarks = collegesDao.collegeJSON;
            JSONArray examBookmarks = collegesDao.examJSON;

request.setAttribute("collegeBookmarks",collegeBookmarks.toString());

request.setAttribute("examBookmarks",examBookmarks.toString());

request.getRequestDispatcher("bookmarks.jsp").forward(request, response);
        } else {
            request.setAttribute("error","true");
            request.getRequestDispatcher("colleges.jsp").forward(request,
response);
        }
    } catch (ClassNotFoundException e) {
        e.printStackTrace();
    }
}
}

```

BookmarksServlet.java

```
package web;

import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import org.json.simple.JSONArray;

import utils.BookmarksUtil;
import db.BookmarksDAO;

@WebServlet("/bookmarks")
public class BookmarksServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;
    private BookmarksDAO bookmarksDao;

    public void init() {
        bookmarksDao = new BookmarksDAO();
    }

    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

        String username = (String) request.getSession(false).getAttribute("email");
        System.out.println("EMAIL ID from Sessions Storage: "+username);

        BookmarksUtil bookmarksUtil = new BookmarksUtil();
        bookmarksUtil.setUsername(username);

        try {
            if (bookmarksDao.validate(bookmarksUtil)) {
                response.setStatus(200);
                JSONArray collegeBookmarks = bookmarksDao.collegeJSON;
                JSONArray examBookmarks = bookmarksDao.examJSON;
            }
        }
    }
}
```

```

request.setAttribute("collegeBookmarks",collegeBookmarks.toString());

request.setAttribute("examBookmarks",examBookmarks.toString());

request.getRequestDispatcher("bookmarks.jsp").forward(request, response);
        } else {
            response.setStatus(401);
        }
    } catch (ClassNotFoundException e) {
        e.printStackTrace();
    }
}
}

```

CODE SNIPPETS FOR JSP:

login.jsp

```

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!--
    Copyright (c) 2020 | Exam Experts
-->
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width, initial-scale=1">
        <title>Exam Experts</title>
        <link rel="stylesheet" href="/css/bulma.min.css">
        <link rel="stylesheet" href="/css/main.css">
        <link
href="https://fonts.googleapis.com/css2?family=Lato&family=Pacifico&family=Poppins&fa
mily=Quicksand&display=swap" rel="stylesheet">
        </head>
        <body>
            <nav class="navbar is-transparent is-fixed-top" role="navigation" aria-label="main
navigation" id="navbar">
                <div class="navbar-menu is-active">
                    <div class="navbar-start">

```

```

        <!-- navbar items -->
        <div class="navbar-item" id="navtext"></div>
    </div>
</div>
</nav>

<section class="section" id="section">
    <div class="columns">
        <div class="column is-6">
            
        </div>
        <div class="column" id="login-form">
            <% if(request.getAttribute("error")=="true"){ %>
                <article class="message"><div class="message-header">Please provide the correct
credentials</div></article>
            <% } %>
            <div class="welcometext has-text-centered">
                Welcome Back
            </div>
            <form action="<%=request.getContextPath()%>/login" id="myform" method="post">
                <div class="field">
                    <label class="label">Email</label>
                    <div class="control">
                        <input class="input" type="email" id="email" name="email"
placeholder="capturesmk@gmail.com">
                    </div>
                </div>

                <div class="field">
                    <label class="label">Password</label>
                    <div class="control">
                        <input class="input" type="password" id="pass" name="pass"
placeholder="secret">
                    </div>
                </div>

                <div class="field">
                    <div class="control">
                        <input id="login-btn" type="submit" class="button is-success is-rounded"
value="Login">
                    </div>
                </div>
            </form>

```

```

        <div class="buttons is-centered">
            <a class="button is-info is-rounded" id="register-btn"
href="<%=request.getContextPath()%>/register">New User? Register Here</a>
        </div>

    </div>
</div>
</section>

<!--Scripts ARENA-->
<script src="https://unpkg.com/typewriter-effect@latest/dist/core.js"></script>
<script src = "https://ajax.googleapis.com/ajax/libs/jquery/2.1.1/jquery.min.js"></script>
<script>
    var app = document.getElementById('navtext');
    var typewriter = new Typewriter(app, {
        loop: true
    });
    typewriter.typeString('Exam Experts')
        .start();
</script>
<!--Scripts ARENA Ends-->
</body>
</html>

```

exams.jsp

```

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<%@ page import="org.json.simple.JSONArray" %>
<%@ page import="org.json.simple.JSONObject" %>
<%@ page import="org.json.simple.parser.JSONParser" %>
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width, initial-scale=1">
        <title>Exam Experts</title>
        <link rel="stylesheet" href="/css/bulma.min.css">
        <link rel="stylesheet" href="/css/main.css">
        <link
href="https://fonts.googleapis.com/css2?family=Lato&family=Pacifico&family=Poppins&fa
mily=Quicksand&display=swap" rel="stylesheet">

```

```
</head>
```

```
<nav class="navbar is-transparent is-fixed-top" role="navigation" aria-label="main
navigation" id="navbar">
```

```
<div class="navbar-menu is-active">
```

```
<div class="navbar-start">
```

```
<!-- navbar items -->
```

```
<div class="navbar-item" id="navtext"></div>
```

```
</div>
```

```
<div class="navbar-end">
```

```
<div class="navbar-item">
```

```
<a href="" id="logout">Logout</a>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</nav>
```

```
<section class="section" id="section">
```

```
<div class="mainhead has-text-centered"><div class="box">Exams</div></div>
```

```
<div id="exam">
```

```
<% String exams = (String) request.getAttribute("exams");
```

```
JSONParser jsonParser = new JSONParser();
```

```
JSONArray jsonArray = (JSONArray)jsonParser.parse(exams);
```

```
System.out.println(jsonArray.size());
```

```
for(int i = 0; i < jsonArray.size(); ++i) {
```

```
    JSONObject jsonObject = (JSONObject) jsonArray.get(i); %>
```

```
<div class="columns is-vcentered" id="exam-list">
```

```
<div class="column side-align">
```

```
<div class="box">
```

```
<form action="<%=request.getContextPath()%>/exams"
```

```
method="post">
```

```
<div class="field">
```

```
<div class="control">
```

```
<input name="eid" type="hidden" value="<%
```

```
out.print(jsonObject.get("ExamID")); %>">
```

```
<div class="buttons is-right clg-btns">
```

```
<input type="submit" class="button is-success is-rounded"
```

```
value="Save to Bookmarks">
```

```
</div>
```

```
</div>
```

```
</div>
```

```

        <div class="exam-name">
        <% out.println(jsonObject.get("ExamName")); %>
        </div>
        <p class="exam-desc">
        <% out.println(jsonObject.get("Description")); %>
        </p>
        <div class="exam-details">
            <span class="tag is-medium is-warning
desc-tags">TotalMarks</span>
            <span class="exam-cutoff"><%
out.println(jsonObject.get("Totalmarks")); %></span>
        </div>
    </div>
</div>
</div>
<% } %>
</div>

</section>
<!--Scripts ARENA-->
<script src="https://unpkg.com/typewriter-effect@latest/dist/core.js"></script>
<script src = "https://ajax.googleapis.com/ajax/libs/jquery/2.1.1/jquery.min.js"></script>
<script>
    var app = document.getElementById('navtext');
    var typewriter = new Typewriter(app, {
        loop: true
    });
    typewriter.typeString('Exam Experts')
        .start();

</script>
<!--Scripts ARENA Ends-->
</body>
</html>

```

colleges.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<%@ page import="org.json.simple.JSONArray" %>
<%@ page import="org.json.simple.JSONObject" %>
<%@ page import="org.json.simple.parser.JSONParser" %>
<!--
    Copyright (c) 2020 | Exam Experts
-->
<html>
    <head>
        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width, initial-scale=1">
        <title>Exam Experts</title>
        <link rel="stylesheet" href="/css/bulma.min.css">
        <link rel="stylesheet" href="/css/main.css">
        <link
href="https://fonts.googleapis.com/css2?family=Lato&family=Pacifico&family=Poppins&fa
mily=Quicksand&display=swap" rel="stylesheet">
        </head>

        <body onload="fetchXML()">
            <nav class="navbar is-transparent is-fixed-top" role="navigation" aria-label="main
navigation" id="navbar">
                <div class="navbar-menu is-active">
                    <div class="navbar-start">
                        <!-- navbar items -->
                        <div class="navbar-item" id="navtext"></div>
                    </div>
                    <div class="navbar-end">
                        <div class="navbar-item">
                            <a href="" id="logout">Logout</a>
                        </div>
                    </div>
                </div>
            </nav>
            <section class="section" id="section">

                <div id="clgpage">
                    <% JSONParser jsonParser = new JSONParser();
                    String college = (String) request.getAttribute("college");
                    JSONArray jsonArray = (JSONArray) jsonParser.parse(college);
```



```

for(int i = 0; i < jsonArray.size(); ++i) {
    JSONObject jsonObject = (JSONObject) jsonArray.get(i); %>
    <div class="columns is-vcentered" id="college-list">
        <div class="column is-2">
            ">
        </div>
        <div class="column side-align">
            <div class="box">
                <form action="<%=request.getContextPath()%>/colleges"
method="post">

                    <div class="field">
                        <div class="control">
                            <input name="uid" type="hidden" value="<%
out.print(jsonObject.get("UniversityID")); %>">
                            <input type="submit" class="button is-success is-rounded"
value="Save to Bookmarks">
                        </div>
                    </div>

                    </form>
                    <div class="clg-name">
                        <% out.println(jsonObject.get("UniversityName")); %>
                    </div>
                    <p class="clg-desc">
                        <% out.println(jsonObject.get("Description")); %>
                    </p>
                    <div class="clg-details">
                        <div class="space-tags">
                            <span class="tag is-medium is-warning
desc-tags">Examination</span>
                                <span class="clg-exam"><%
out.println(jsonObject.get("ExamID")); %></span>
                            </div>
                        <div>
                            <span class="tag is-medium is-warning
desc-tags">Cut-Off Score</span>
                                <span class="clg-cutoff"><%
out.println(jsonObject.get("Cutoff")); %></span>
                            </div>
                        </div>
                    <div class="buttons is-right clg-btns">

```

```

        <a class="button is-dark is-rounded clg-location"
target="_blank" href="<% out.println(jsonObject.get("Location")); %>">Locate in Maps</a>
        <a class="button is-link clg-url" target="_blank" href="<%
out.println(jsonObject.get("URL")); %>">View More</a>
    </div>
</div>
</div>
<%} %>
</div>

```

```

<!--Scripts ARENA-->
<script src="https://unpkg.com/typewriter-effect@latest/dist/core.js"></script>
<script src = "https://ajax.googleapis.com/ajax/libs/jquery/2.1.1/jquery.min.js"></script>
<script>
    var app = document.getElementById('navtext');
    var typewriter = new Typewriter(app, {
        loop: true
    });
    typewriter.typeString('Exam Experts')
        .start();

    // LOADING XML RESPONSE

</script>
<!--Scripts ARENA Ends-->
</body>
</html>

```

bookmarks.jsp

```

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<%@ page import="org.json.simple.JSONArray" %>
<%@ page import="org.json.simple.JSONObject" %>
<%@ page import="org.json.simple.parser.JSONParser" %>
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Exam Experts</title>
    <link rel="stylesheet" href="./css/bulma.min.css">

```



```

        <div class="space-tags">
            <span class="tag is-medium is-warning
desc-tags">Examination</span>
            <span class="clg-exam"><%
out.println(jsonObject.get("ExamID")); %></span>
        </div>
        <div>
            <span class="tag is-medium is-warning
desc-tags">Cut-Off Score</span>
            <span class="clg-cutoff"><%
out.println(jsonObject.get("Cutoff")); %></span>
        </div>
    </div>
    <div class="buttons is-right clg-btns">
        <a class="button is-dark is-rounded clg-location"
target="_blank" href="<% out.println(jsonObject.get("Location")); %>">Locate in Maps</a>
        <a class="button is-link clg-url" target="_blank" href="<%
out.println(jsonObject.get("URL")); %>">View More</a>
    </div>
</div>
</div>
<%} %>
</div>
<div class="mainhead has-text-centered"><div class="box">Bookmarked
Exams</div></div>
<div id="exambk">
<% String examBookmarks = (String) request.getAttribute("examBookmarks");
JSONArray jsonArray2 = (JSONArray) jsonParser.parse(examBookmarks);
for(int i = 0; i < jsonArray2.size(); ++i) {
    JSONObject jsonObject2 = (JSONObject) jsonArray2.get(i); %>
    <div class="columns is-vcentered" id="exam-list">
        <div class="column side-align">
            <div class="box">
                <div class="exam-name">
                    <% out.println(jsonObject2.get("ExamName")); %>
                </div>
                <p class="exam-desc">
                    <% out.println(jsonObject2.get("Description")); %>
                </p>
                <div class="exam-details">
                    <span class="tag is-medium is-warning
desc-tags">Total Marks</span>

```

```

                                <span class="exam-cutoff"><%
out.println(jsonObject2.get("Totalmarks")); %></span>
                                </div>
                                </div>
                                </div>
                                </div>
                                <% } %>
                                </div>
                                </section>
                                <script src="https://unpkg.com/typewriter-effect@latest/dist/core.js"></script>
                                <script src =
                                "https://ajax.googleapis.com/ajax/libs/jquery/2.1.1/jquery.min.js"></script>
                                <script>
                                var app = document.getElementById('navtext');
                                var typewriter = new Typewriter(app, {
                                loop: true
                                });
                                typewriter.typeString('Exam Experts')
                                .start();
                                </script>
                                </body>
                                </html>

```

VALIDATION

LOGIN

Username	Password	Output
sai.sanjay7@gmail.com	Saisan27	VALID
capturesmk@gmail.com		INVALID
msmuthu.sk@gmail.com	msm	INVALID
ajaypranav06@gmail.com	Ajaypran06	VALID

Output is returned as **INVALID** where the email or/and password are incorrect or NULL.

EXAMS

ExamID	ExamName	Description	Totalmarks	Output
GRE	Graduate Record Examination	The Graduate Record Examinations is a standardized test that is an admissions requirement for many graduate schools in the United States and Canada.	340	VALID
GATE	Gate	The Graduate Aptitude Test in Engineering (GATE) is an examination that primarily tests the comprehensive understanding of	100	INVALID

		various undergraduate subjects in engineering		
IELTS	International English Language Testing System		9	INVALID
TOEFL	Test of English as a Foreign Language	Test of English as a Foreign Language is a standardized test to measure the English language ability of non-native speakers wishing to enroll in English-speaking universities	120	VALID

Output is returned as **INVALID** if any of the properties have a value not present in the database or NULL.

BOOKMARKS

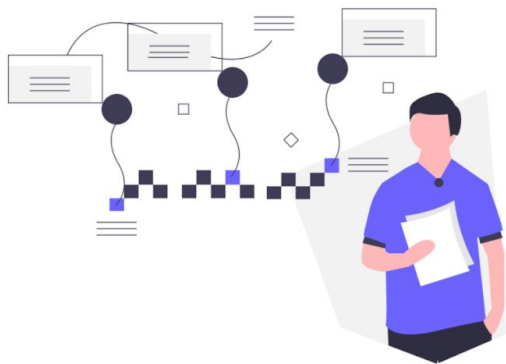
EmailID	ExamID	UniversityID	Output
sai.sanjay7@gmail.com	TOEFL	LMU	VALID
capturesmk@gmail.com		IIT	VALID
msmuthu.sk@gmail.com	RMA		INVALID
ajaypran06			INVALID

Output is returned as **INVALID** if any of the properties have a value not present in the database or NULL.

SCREENSHOTS

LOGIN PAGE

Exam Expert



Welcome Back

Email

sai.sanjay7@gmail.com

Password

New User? Register Here

Login

DASHBOARD

Exam Ex

Logout

My Dashboard



My Bookmarks

Find your Saved Bookmarks here



Colleges

Search exams to enter your desired college



Exams

Search your desired exams in one go.



Rank Predictor

Predict your results with our data

EXAMS

Exam Explorer

Logout

Exams

Graduate Record Examination

The Graduate Record Examinations is a standardized test that is an admissions requirement for many graduate schools in the United States and Canada.

549

Save to Bookmarks

Graduate Aptitude Test in Engineering

The Graduate Aptitude Test in Engineering (GATE) is an examination that primarily tests the comprehensive understanding of various undergraduate subjects in engineering

100

Save to Bookmarks

International English Language Testing System

The International English Language Testing System, or IELTS, is an international standardized test of English language proficiency for non-native English language speakers

9

Save to Bookmarks

Test of English as a Foreign Language

Test of English as a Foreign Language is a standardized test to measure the English language ability of non-native speakers wishing to enroll in English-speaking universities

120

Save to Bookmarks

COLLEGES

Exam Explorer


Logout

Filters

Masters of Technology

United States Of America

Apply Filter



Northern University

Save to Bookmarks


Northeastern University (NU or NEU) is a private research university in Boston, Massachusetts, established in 1898. The university offers undergraduate and graduate programs on its main campus in Boston.

Examination GRE

Cut-Off Score 299

Locate in Maps

View More



Indian Institute of Technology

Save to Bookmarks

Indian Institute of Technology is one of the top universities created to be Centres of Excellence for training, research and development in science, engineering and technology in India.

Examination GATE

BOOKMARKS

Exam Experts/

[Logout](#)

Bookmarked Colleges



London Metropolitan University

London Metropolitan University, commonly known as London Met, is a public research university in London, England. The University of North London and London Guildhall University merged in 2002 to create the university

Examination	IELTS
Cut-Off Score	6

[Locate in Maps](#) [View More](#)

Bookmarked Exams

Graduate Record Examination

The Graduate Record Examinations is a standardized test that is an admissions requirement for many graduate schools in the United States and Canada.

Total Marks	340
-------------	-----