Course-Registration-Online-System for Controller

CourseRegisterServlet.java

- The @WebServlet annotation is used to declare a servlet with parameter(“\courseRegister”).
- The annotated class must extend the HttpServlet class, so CourseRegisterServlet must extend HttpServlet class.
- The @PostConstruct annotation is used on a method that needs to be executed after dependency injection is done to perform any initialization.
- CourseRegisterServlet class has doPost method with parameters HttpServletRequest request, HttpServletResponse response and throws ServletException and IOExeception.
- Class check with if (null==) condition and if equal with null it will do response sendRedirect with parameter request for contextPath for index.jsp else do requestParameter for value and response sendRedirect with parameter request for contextPath next file confirm.jsp.

UserLoginServlet.java

- The @WebServlet annotation is used to declare a servlet with parameter(“user\login”).
- The annotated class must extend the HttpServlet class, so UserLoginServlet must extend HttpServlet class.
- The @PostConstruct annotation is used on a method that needs to be executed after dependency injection is done to perform any initialization.

- The @Override annotation indicates that the child class method is over-writing its base class method.
- UserLoginServlet has doGet() method is called by the server to allow a servlet to handle a GET request and use doGet() method for getting the information from the server. CourseRegisterServlet class has doGet method with parameters HttpServletRequest request, HttpServletResponse response and throws ServletException and IOExeception.
- Within the doPost method has getParameter for value and getSession(), sendRedirect with parameter getContextPath() for dashboard.jsp.

UserLogoutServlet.java

- The @WebServlet annotation is used to declare a servlet with parameter(“user\logout”).
- The annotated class must extend the HttpServlet class, so UserLogoutServlet must extend HttpServlet class.
- The @Override annotation indicates that the child class method is over-writing its base class method.
- UserLogoutServlet has doGet() method is called by the server to allow a servlet to handle a GET request and use doGet() method for getting the information from the server. UserLogoutServlet class has doGet method with parameters HttpServletRequest request, HttpServletResponse response and throws ServletException and IOExeception.
- The doGet() method has req.getSession().invalidate(); and resp.sendRedirect(req.getContextPath()); methods.

UserRegisterServlet.java

- The @WebServlet annotation is used to declare a servlet with parameter(“users\register”).
- The annotated class must extend the HttpServlet class, so UserRegisterServlet must extend HttpServlet class.
- The @PostConstruct annotation is used on a method that needs to be executed after dependency injection is done to perform any initialization**.**
- The doPost method parameter with parameters HttpServletRequest request, HttpServletResponse response and throws ServletException and IOExeception.
- Within doPost method has req.getParameter for to get value resp.sendRedirect(req.getContextPath() + "/users") .

UserServlet.java

- The @WebServlet annotation is used to declare a servlet with parameter(“/users”).
- The annotated class must extend the HttpServlet class, so the class UserServlet must extend HttpServlet class.
- The @PostConstruct annotation is used on a method that needs to be executed after dependency injection is done to perform any initialization.
- The @Override annotation indicates that the child class method is over-writing its base class method.
- The class UserServlet has doGet Method with parameter HttpServletRequest request, HttpServletResponse response and throws ServletException and IOExeception.

- The class also include an Iterator and it is an object that can be used to loop through collections. If we use iterator, we must import it from the java.

Course-Registration-Online-System for Entity

Course.java

- Declare variables within Course class and also have getter, setter method for each variable.
- The @Override annotation indicates that the child class method is over-writing its base class method and there has toString() return string.

Department.java

- Declare variables within Department class and also have getter, setter method for each variable.

Registration.java

- Declare variables within Department class and also have getter, setter method for each variable.

Student.java

- Declare variables within Student class and also have getter, setter method for each variable.
- The @Override annotation indicates that the child class method is over-writing its base class method and there has toString() return string.

Course-Registration-Online-System for Repository

CourseRepository.java

- The CourseRepository class build object for ArrayList and set values for course and also include getter setter method.
- The getById method with parameter String id and use for loop to return courses.
- Method List<Course> findByDepartment with parameter (Department department) and object for arraylist and the next line use for loop to get department and return list.

DepartmentRepository.java

- The DepartmentRepository class build object for Linklist and within static do set department info, history of department and add values.
- Department findById method use for loop to get department with id and also include throw exception if department id is not found.
- This class has getter and setter method for department.

RegistrationRepository.java

- The RegistrationRepository class build object for Linklist and build constructor class.
- The next class call object and use for loop to get student with id and add registration.

- Registration getById method has for loop to get registration with id and also has throw exception for not found case and then also has setmethod for registration.

StudentRepository.java

- The StudentRepository class build object for Linklist and within static class set values for student and also has getter, setter method for student.
- Student getUserById method with parameter id and has for loop to get student with id and also include throw exception for not found.
- Student getUserByEmail method with String parameter and email to change lowercase and then the next line use for loop with student size to get student info.

Course-Registration-Online-System for Filter

CourseRegisterFilter.java

- The @WebFilter annotation is used to define a Filter in a web application and with parameter(“\courseRegister”).
- The CourseRegister class implements filer and within class use @override annotation.
- This class has doFilter method with (ServletRequest request, ServletResponse response, FilterChain chain) parameter and throws IOException, ServletException.
- If condition with getParameterMap method to get value and throw new NoSuchElementException for not found case.

- If condition to check isEmpty() with OR operator and if it true ,will do doFilter method with (request, response) parameter.

RegisterFilter.java

- The @WebFilter annotation is used to define a Filter in a web application and with parameter(“/users/register”).
- The RegisterFilter class implements filer and within class use @override annotation.
- This class has doFilter method with (ServletRequest request, ServletResponse response, FilterChain chain) parameter and throws IOException, ServletException.
- If condition with getParameterMap method to get value and throw new NoSuchElementException for not found case.
- Declare variables to assign value from object. getParameter (“variable”). toString(); .
- If condition to check student info with OR operator and throw new IllegalArgumentException for not found case.
- If condition to check email and password and if true do doFilter method with (request, response) parameter.

StudentFilter.java

- The @WebFilter annotation is used to define a Filter in a web application and with parameter(“/users/login”).
- The RegisterFilter class implements filer and within class use @override annotation.

- This class has doFilter method with (ServletRequest request, ServletResponse response, FilterChain chain) parameter and throws IOException, ServletException.
- If condition with getParameterMap method to get value and throw new NoSuchElementException for not found case.
- Declare variable to assign value from object. getParameter (“variable”). toString().
- If condition to check email and password with OR operator and throw new IllegalArgumentException for empty.
- If email match do doFilter method.

Course-Registration-Online-System for Exception

DepartmentNotFound.java

- Class DepartmentNotFound extends RuntimeException and within class has DepartmentNotFound class with string variable parameter and with these class has super method.

IncorrectPasswordException.java

- Class IncorrectPasswordException extends RuntimeException and within class has IncorrectPasswordException class with string variable parameter and with these class has super method.

InvalidEmailAddressException.java

- Class InvalidEmailAddressExeption extends RuntimeException and with class has InvalidEmailAddressException class with string variable parameter and with these class has super method.

UserNotFoundException.java

- Class UserNotFound extends RuntimeException and within class has UserNotFound class with string variable and with these class has super method.

Course-Registration-Online-System for Service

CourseService.java

- The class CourseService declare private variable and has constructor within these build object.
- Use arraylist to get department and the next is getCourseByDepartment use try catch exception and return arraylist.
- Also include getCourse with arraylist and setCourse methods.

DepartmentService.java

- The class DepartmentService declare variable for repository and build constructor.
- Within constructor new object assign to variable and the next class return getDepartment with arraylist.

RegistrationService.java

- The class RegistrationService declare variable repository and build constructor.
- Within constructor new object assign to variable and the next return getRegistrationById with arraylist.
- Save method with parameter set values.

StudentService.java

- The class StudentService class declare private variable for repository and build constructor, within constructor variable assign to new object.
- Getusers with arraylist and also include setuser method.
- The register method with parameters and set values.
- Login method with email, password parameter and if condition to check password and throw new exception for incorrect password case and return student.

Course-Registration-Online-System for Jsp

confirm.jsp

- Page import com.its20.demo.entity.Student com.its20.demo.service.RegistrationService com.its20.demo.entity.Registration com.its20.demo.service.CourseService.
- Use if condition to check user id and if it true return index.jsp else if request student.

dashboard.jsp

Page import com.its20.demo.entity.Course, com.its20.demo.service.CourseService, com.its20.demo.entity.Department, com.its20.demo.service.DepartmentService, com.its20.demo.entity.Student.

- Use if to check user attribute is null and return index.js and else if get attribute of user with session.

index.jsp

- Check with if condition user attribute is null send direct response to dashboard.jsp.

register.jsp

- Page import com.its20.demo.entity.Registration, com.its20.demo.service.RegistrationService and com.its20.demo.entity.Student.
- Check with if condition user attribute is null return index.jsp and else if get user attribute with session.