MỤC TIÊU:

Kết thúc bài thực hành này bạn có khả năng

- …

YÊU CẦU:

Xây dựng giải pháp security cho Assignment của môn học này

PHẦN I

BÀI 1 (2 ĐIỂM)

Tạo một dự án web động bao gồm các thành phần sau:

- ✓ User: thực thể ánh xạ với bảng Users
- ✓ Interface UserDAO, class UserDAOImpl: truy xuất dữ liệu
- ✓ Khai báo các thư viện phụ thuộc cần thiết (DB Driver, HibernateJPA)
- ✓ Tạo các servlet cần có để chạy thử
  - @WebServlet({"/account/sign-up", "/account/change-password", "/account/edit-profile"}): Quản lý tài khoản
  - @WebServlet({"/video/list", "/video/detail/*", "/video/like/*", "/video/share/*"}): xem, like, share video
  - @WebServlet({"/admin/video", "/admin/user", "/admin/like", "/admin/share"}): Quản trị dữ liệu
- ✓ page.jsp: trang jsp có nội dung "Xin chào" để làm trang chung cho tất cả các servlet trên chuyển đến khi nhận request từ người dùng.

BÀI 2 (2 ĐIỂM)

Xây dựng bộ lọc AuthFilter để lọc các request đến các servlet *{"/admin/*", "/account/change-password", "/account/edit-profile", "/video/like/*", "/video/share/*"}* và thực hiện viết mã cho doFilter() để thực hiện phân quyền sử dụng theo yêu cầu sau:

- ✓ Truy xuất đến tất cả URL trên: yêu cầu đăng nhập
- ✓ Truy xuất "/admin/*": yêu cầu đăng nhập với vai trò admin

Hướng dẫn:

✓ Tạo poly.filter.AuthFilter có tổ chức như sau

```
@WebFilter({
        "/admin/*",
        "/account/change-password",
        "/account/edit-profile",
        "/video/like/*",
        "/video/share/*"
})
public class AuthFilter implements Filter {
        public static final String SECURITY_URI = "securityUri";
        @Override
        public void destroy() { }
        @Override
        public void doFilter(ServletRequest request, ServletResponse response, FilterChain chain)
                        throws IOException, ServletException {
                ...
        }
        @Override
        public void init(FilterConfig filterConfig) throws ServletException { }
}
```

✓ Cài đặt mã cho phương thức doFilter() để hiện thực security theo yêu cầu

```
@Override
public void doFilter(ServletRequest request, ServletResponse response, FilterChain chain)
                throws IOException, ServletException {
        HttpServletRequest req = (HttpServletRequest) request;
        HttpServletResponse resp = (HttpServletResponse) response;
        HttpSession session = req.getSession();
        User user = (User) session.getAttribute("user");
        String uri = req.getRequestURI();
        if (user == null || (uri.contains("/admin/") && !user.getAdmin())) {
                session.setAttribute(AuthFilter.SECURITY_URI, uri);
                resp.sendRedirect(req.getContextPath() + "/login");
        } else {
                chain.doFilter(request, response);
        }
}
```

PHẦN II

BÀI 3 (2 ĐIỂM)

Xây dựng chức năng đăng nhập. Nếu đăng nhập thành công thì quay lại trang được bảo mật được yêu cầu lúc chưa đăng nhập.

Hướng dẫn:

- ✓ Tạo giao diện trang đăng nhập (login.jsp)

```
<c:url var="url" value="/login"></c:url>
<i>${message}</i>
<form action="${url}" method="post">
        <input name="username"><br>
        <input name="password" type="password"><hr>
        <button>Login</button>
</form>
```

- ✓ Tạo Servlet điều khiển đăng nhập

```
@WebServlet("/login")
public class LoginServlet extends HttpServlet {
        @Override
        protected void doGet(HttpServletRequest req, HttpServletResponse resp)
                        throws ServletException, IOException {
                req.getRequestDispatcher("/login.jsp").forward(req, resp);
        }

        @Override
        protected void doPost(HttpServletRequest req, HttpServletResponse resp)
                        throws ServletException, IOException {
                ....
        }
}
```

- ✓ Cài đặt mã nguồn cho phương thức doPost()

```
String username = req.getParameter("username");
String password = req.getParameter("password");

UserDAO dao = new UserDAOImpl();
User user = dao.findById(username);
if (user == null) {
        req.setAttribute("message", "Invalid username");
} else if (!user.getPassword().equals(password)) {
        req.setAttribute("message", "Invalid password");
} else {
        HttpSession session = req.getSession();
        session.setAttribute("user", user);
        req.setAttribute("message", "Login successfully");

        String securityUri = (String) session.getAttribute(AuthFilter.SECURITY_URI);
        if (securityUri != null) {
                resp.sendRedirect(securityUri);
                return;
        }
}
req.getRequestDispatcher("/login.jsp").forward(req, resp);
```

BÀI 4 (2 ĐIỂM)

Hãy chạy thử với các địa chỉ url sau đây mỗi vai trò 1 lần.

- ✓ Trang không được bảo vệ:
  - /account/sign-in
- ✓ Trang yêu cầu đăng nhập:
  - /account/change-password
  - /account/edit-profile
  - /video/like/*
  - /video/share/*
- ✓ Trang yêu cầu đăng nhập với vai trò admin
  - /admin/video
  - /admin/user
  - /admin/like
  - /admin/share

BÀI 5 GIẢNG VIÊN CHO THÊM (2 ĐIỂM)