

## MỤC TIÊU:

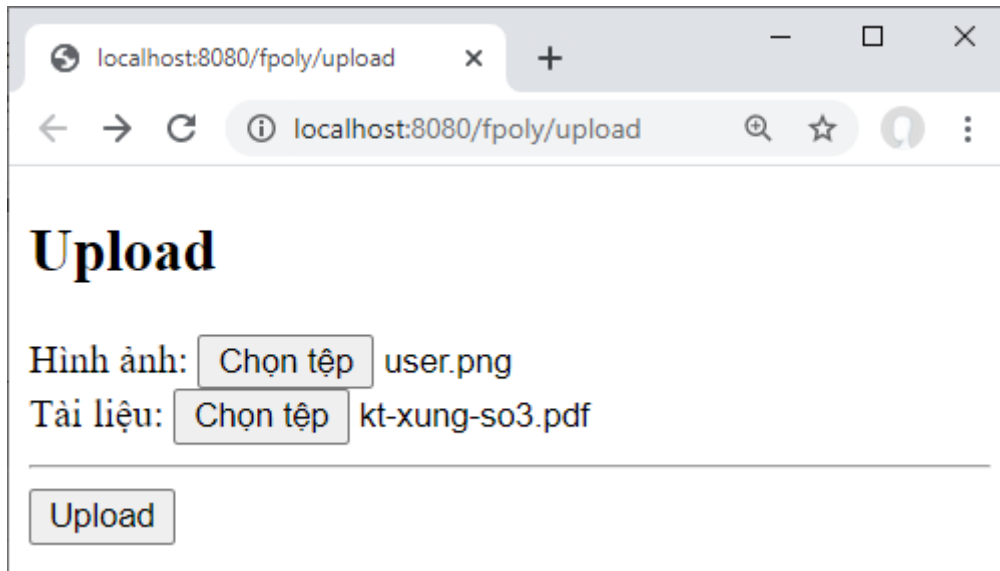
Kết thúc bài thực hành này bạn có khả năng

- Upload file
- Sử dụng BeanUtils
- Sử dụng Cookie
- Gửi email thông qua gmail

## PHẦN I: UPLOAD & BEANUTILS

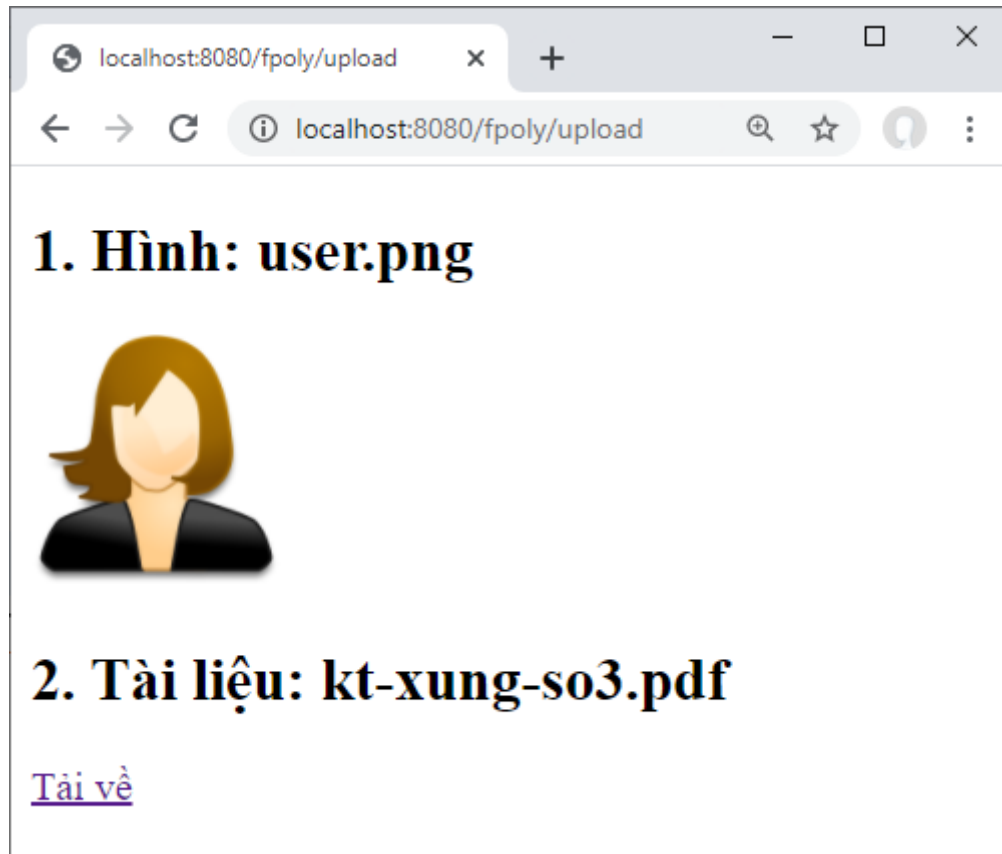
### BÀI 1: UPLOAD FILE

Xây dựng trang web cho phép upload 1 file hình và 1 file tài liệu bất kỳ. Sau khi upload xong thì hiển thị hình đã upload và download trở lại file tài liệu.



The screenshot shows a web browser window with the address bar displaying 'localhost:8080/fpoly/upload'. The page content includes a heading 'Upload', followed by two file selection fields. The first field is labeled 'Hình ảnh:' and shows a selected file 'user.png'. The second field is labeled 'Tài liệu:' and shows a selected file 'kt-xung-so3.pdf'. Below these fields is a button labeled 'Upload'.

Hình 1: Form upload file



Hình 2: Kết quả hiển thị các file đã upload

Hướng dẫn:

- Tạo UploadServlet.java, form.jsp và result.jsp
- Tổ chức UploadServlet.java

```
@MultipartConfig
@WebServlet("/upload")
public class UploadServlet extends HttpServlet{
    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp)
        throws ServletException, IOException {
        req.getRequestDispatcher("/views/upload/form.jsp").forward(req, resp);
    }
    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse resp)
        throws ServletException, IOException {
        // Mã xử lý file upload ở đây
        req.getRequestDispatcher("/views/upload/result.jsp").forward(req, resp);
    }
}
```

```
}  
}
```

- Thiết kế form nhập cho form.jsp

```
<form action="/fpoly/upload" method="post" enctype="multipart/form-data">  
  Hình ảnh: <input type="file" name="photo_file"> <br>  
  Tài liệu: <input type="file" name="doc_file"> <br>  
  <hr>  
  <button>Upload</button>  
</form>
```

- Viết mã xử lý file upload vào doPost() của UploadServlet.java

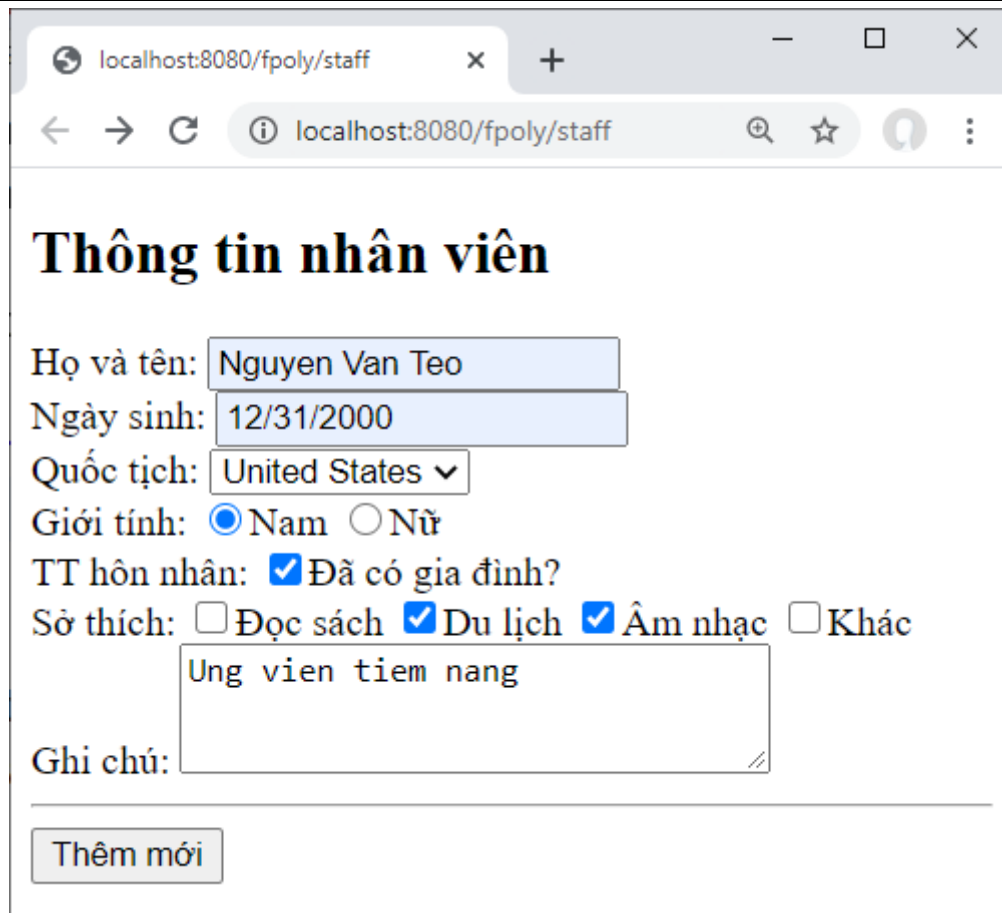
```
// đường dẫn thư mục tính từ gốc của website  
File dir = new File(req.getServletContext().getRealPath("/files"));  
if(!dir.exists()) { // tạo nếu chưa tồn tại  
    dir.mkdirs();  
}  
// lưu các file upload vào thư mục files  
Part photo = req.getPart("photo_file"); // file hình  
File photoFile = new File(dir, photo.getSubmittedFileName());  
photo.write(photoFile.getAbsolutePath());  
Part doc = req.getPart("doc_file"); // file tài liệu  
File docFile = new File(dir, doc.getSubmittedFileName());  
doc.write(docFile.getAbsolutePath());  
  
// chia sẻ cho result.jsp để hiển thị  
req.setAttribute("img", photoFile);  
req.setAttribute("doc", docFile);
```

- Thiết kế trang hiển thị kết quả result.jsp

```
<h2>1. Hình: ${img.name}</h2>  
  
  
<h2>2. Tài liệu: ${doc.name}</h2>  
<a href="/fpoly/files/${doc.name}">Tải về</a>
```

## BÀI 2: SỬ DỤNG BEANUTILS

Hãy sử dụng BeanUtils để đọc tham số form thông tin nhân viên sau đây.



localhost:8080/fpoly/staff

## Thông tin nhân viên

Họ và tên:

Ngày sinh:

Quốc tịch:

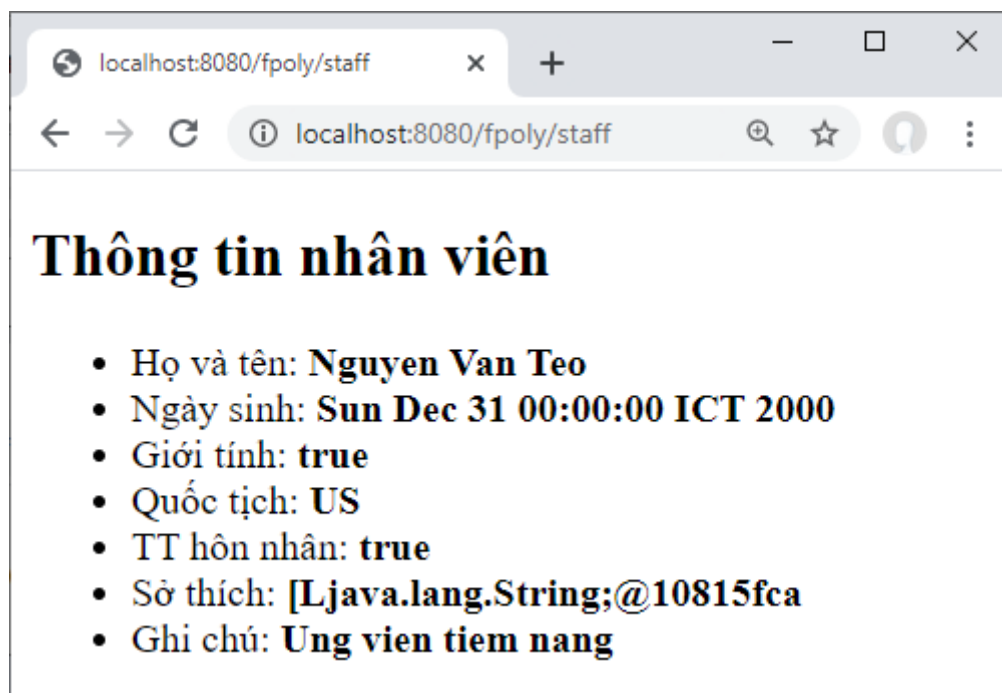
Giới tính: ☒ Nam ☐ Nữ

TT hôn nhân: ☒ Đã có gia đình?

Sở thích: ☐ Đọc sách ☒ Du lịch ☒ Âm nhạc ☐ Khác

Ghi chú:

Hình 1: Form thông tin nhân viên



localhost:8080/fpoly/staff

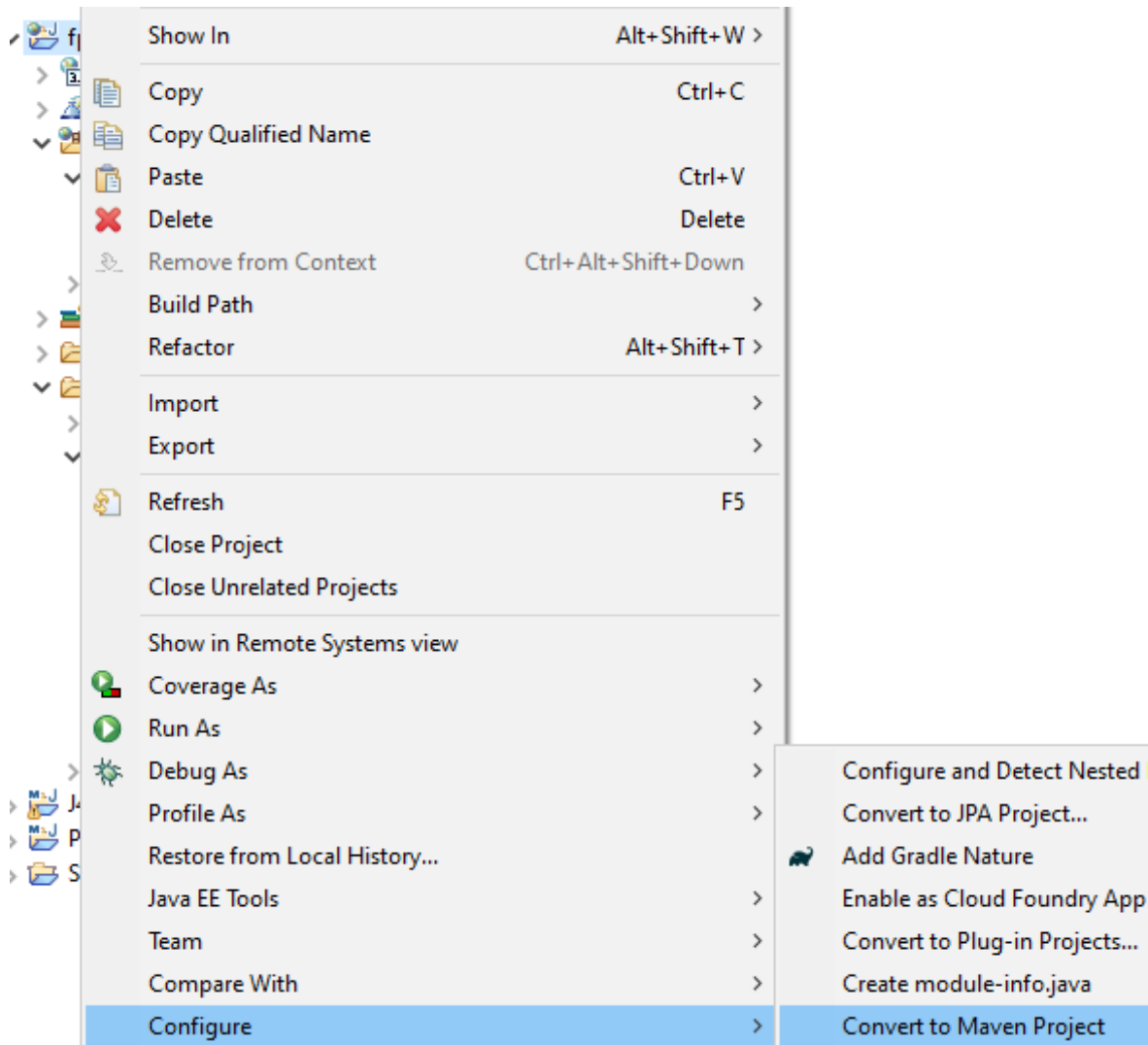
## Thông tin nhân viên

- Họ và tên: **Nguyen Van Teo**
- Ngày sinh: **Sun Dec 31 00:00:00 ICT 2000**
- Giới tính: **true**
- Quốc tịch: **US**
- TT hôn nhân: **true**
- Sở thích: **[Ljava.lang.String;@10815fca**
- Ghi chú: **Ung vien tiem nang**

Hình 2: Thông tin nhân viên hiển thị từ bean

Hướng dẫn:

- Trước khi viết mã cần convert dự án sang Maven và khai báo thư viện BeanUtils vào pom.xml theo hướng dẫn của các hình sau.



Convert project sang Maven

Thông tin về pom.xml

```
<packaging>war</packaging>
<dependencies>
  <!-- BeanUtils -->
  <dependency>
    <groupId>commons-beanutils</groupId>
    <artifactId>commons-beanutils</artifactId>
    <version>1.9.3</version>
  </dependency>
</dependencies>
<build>
```

Khai báo thư viện cần thiết

- Tạo StaffServlet.java, Staff.java, form.jsp và result.jsp
- Tổ chức mã StaffServlet.java

```
@WebServlet("/staff")
public class StaffServlet extends HttpServlet{
    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp)
        throws ServletException, IOException {
        req.getRequestDispatcher("/views/staff/form.jsp").forward(req, resp);
    }
    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse resp)
        throws ServletException, IOException {
        // Mã đọc tham số bằng BeanUtils
        req.getRequestDispatcher("/views/staff/result.jsp").forward(req, resp);
    }
}
```

- Thiết kế form.jsp

```
<form action="/fpoly/staff" method="post">
    Họ và tên: <input name="fullname"> <br>
    Ngày sinh: <input name="birthday"> <br>
    Quốc tịch:
    <select name="country">
        <option value="VN">Việt Nam</option>
        <option value="US">United States</option>
    </select> <br>
    Giới tính:
    <input name="gender" type="radio" value="true"> Nam
    <input name="gender" type="radio" value="false"> Nữ <br>
    TT hôn nhân:
    <input name="married" type="checkbox"> Đã có gia đình? <br>
    Sở thích:
    <input name="hobbies" type="checkbox" value="R"> Đọc sách
    <input name="hobbies" type="checkbox" value="T"> Du lịch
    <input name="hobbies" type="checkbox" value="M"> Âm nhạc
    <input name="hobbies" type="checkbox" value="O"> Khác <br>
    Ghi chú: <textarea name="notes" rows="3" cols="30"> </textarea>
</hr>
```

```
<button>Thêm mới</button>
</form>
```

- Viết mã cho bean Staff.java

```
public class Staff {
    private String fullname;
    private Date birthday;
    private boolean gender;
    private String country;
    private boolean married;
    private String[] hobbies;
    private String notes;

    getters/setters
}
```

- Bổ sung mã cho StaffServlet.java để đọc form bằng bean

```
try {
    // Định dạng thời gian nhập vào
    DateTimeConverter dtc = new DateConverter(new Date());
    dtc.setPattern("MM/dd/yyyy");
    ConvertUtils.register(dtc, Date.class);

    Staff staff = new Staff();
    // Đọc tham số vào các thuộc tính của bean staff
    BeanUtils.populate(staff, req.getParameterMap());

    // Chia sẻ với result.jsp
    req.setAttribute("bean", staff);
} catch (Exception e) {
    e.printStackTrace();
}
```

- Thiết kế result.jsp để hiển thị kết quả

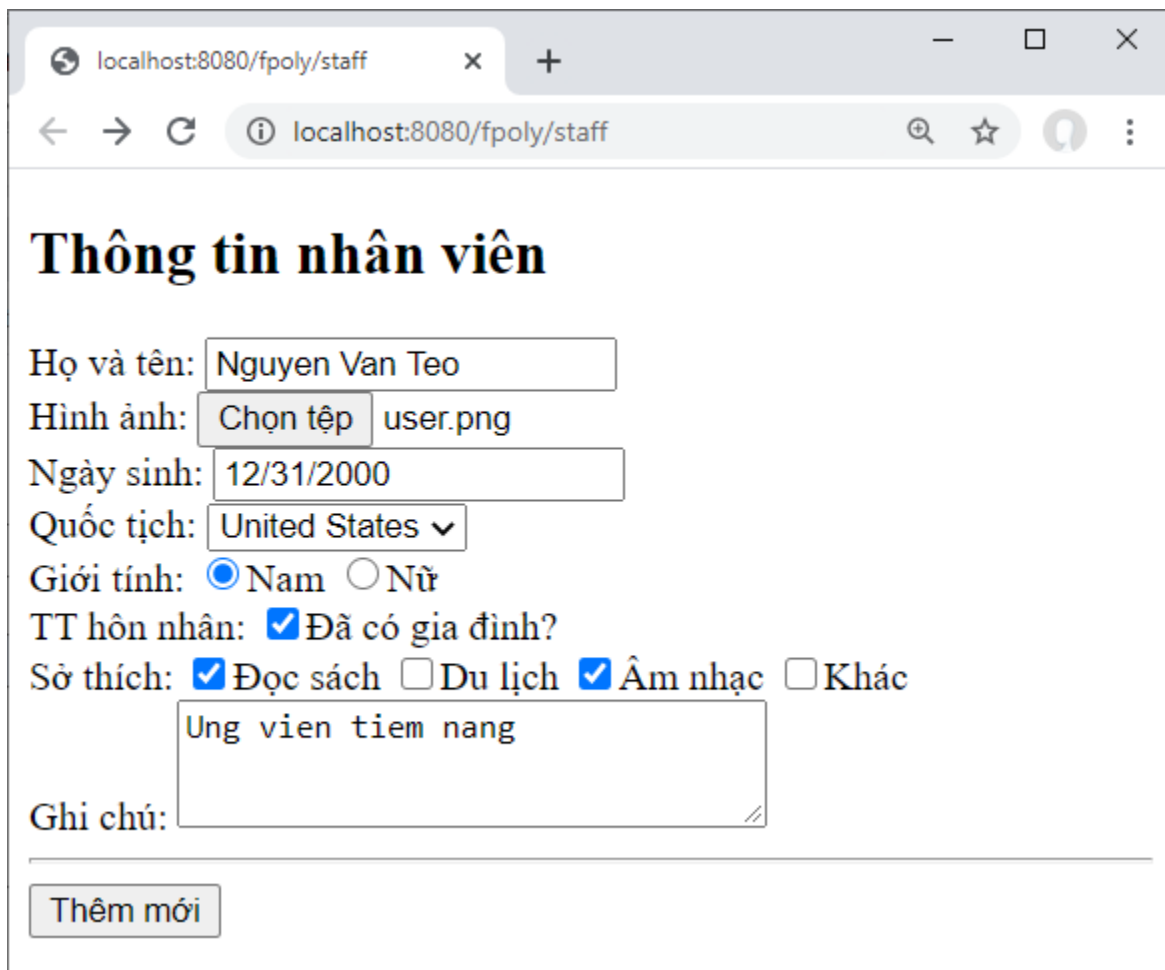
```
<ul>
    <li>Họ và tên: <b>${bean.fullname}</b></li>
    <li>Ngày sinh: <b>${bean.birthday}</b></li>
    <li>Giới tính: <b>${bean.gender}</b></li>
    <li>Quốc tịch: <b>${bean.country}</b></li>
```



```
<li>TT hôn nhân: <b>${bean.married}</b></li>
<li>Sở thích: <b>${bean.hobbies}</b></li>
<li>Ghi chú: <b>${bean.notes}</b></li>
</ul>
```

### BÀI 3: KẾT HỢP UPLOAD VÀ BEANUTILS

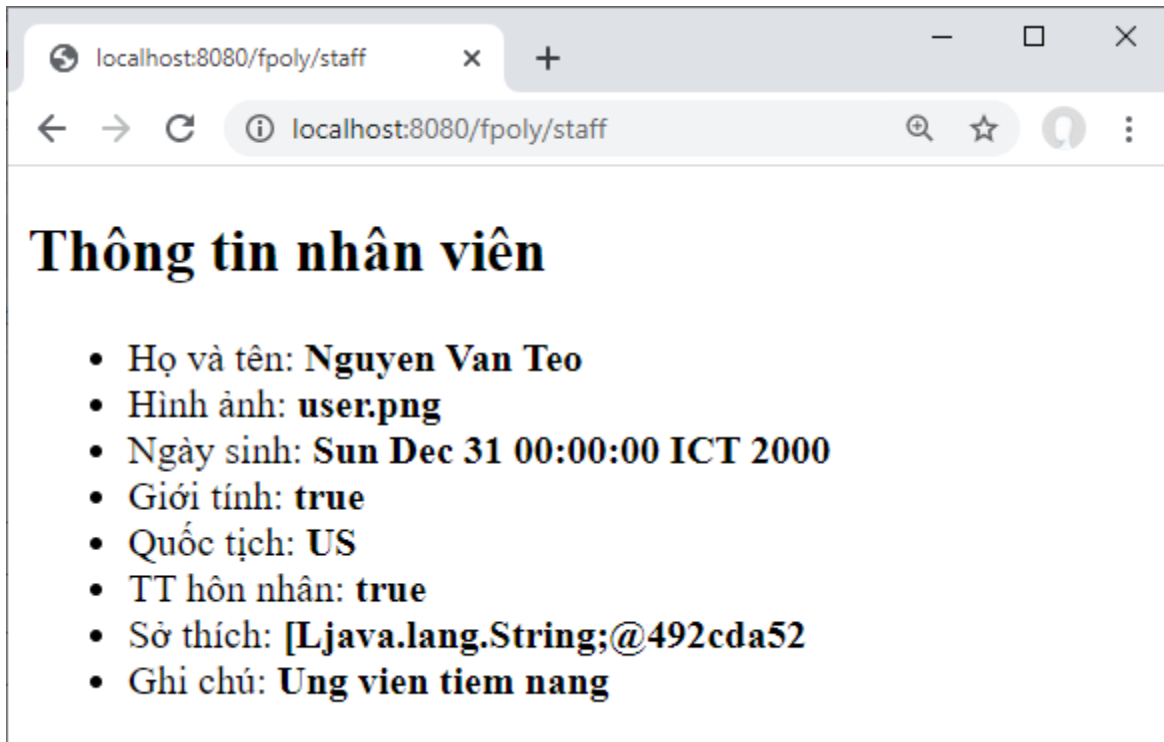
Hãy nâng cấp ứng dụng bài 2 bằng cách thêm hình ảnh nhân viên được mô tả như 2 hình sau.



The screenshot shows a web browser window with the address bar displaying 'localhost:8080/fpoly/staff'. The page title is 'Thông tin nhân viên'. The form contains the following fields and controls:

- Họ và tên: Text input field containing 'Nguyen Van Teo'.
- Hình ảnh: A file selection button labeled 'Chọn tệp' followed by the filename 'user.png'.
- Ngày sinh: Text input field containing '12/31/2000'.
- Quốc tịch: Dropdown menu showing 'United States'.
- Giới tính: Radio buttons for 'Nam' (selected) and 'Nữ'.
- TT hôn nhân: Checkboxes for 'Đã có gia đình?' (selected) and 'Chưa có gia đình?' (unselected).
- Sở thích: Checkboxes for 'Đọc sách' (selected), 'Du lịch' (unselected), 'Âm nhạc' (selected), and 'Khác' (unselected).
- Ghi chú: A large text area containing the text 'Ung viên tiềm năng'.
- Thêm mới: A button at the bottom of the form.

Hình 1: Thông tin nhân viên có hình



Hình 2: Kết quả hiển thị có tên hình

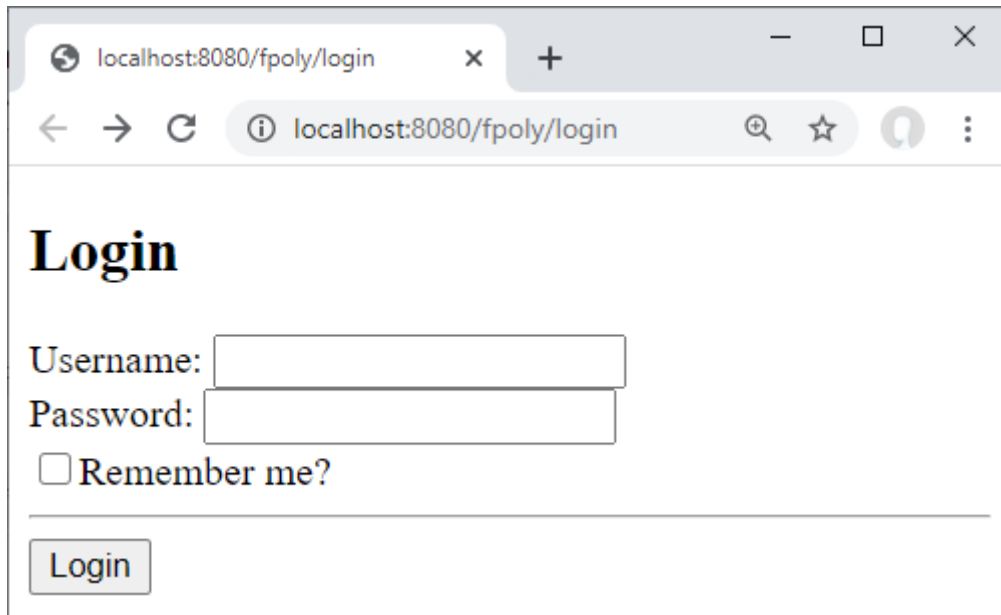
Hướng dẫn:

- Form upload cần
  - Thuộc tính `enctype="multipart/form-data"`
  - `<input name="photo_file" type="file">`
- Bean Staff.java cần thêm
  - `private String photo` và getter/setter
- Servlet StaffServlet.java cần `@MultipartConfig` và bổ sung mã upload trường `photo_file` vào `doPost()`
  - `File dir = new File(req.getServletContext().getRealPath("/files"));`
  - `Part photo = req.getPart("photo_file");`
  - `File photoFile = new File(dir, photo.getSubmittedFileName());`
  - `photo.write(photoFile.getAbsolutePath());`
- Lấy tên file upload đưa vào bean staff
  - `staff.setPhoto(photoFile.getName());`
- Hiển thị kết quả `result.jsp` hiển thị thêm tên photo
  - `<li>Hình ảnh: <b>${bean.photo}</b></li>`

## PHẦN II: COOKIE & EMAIL

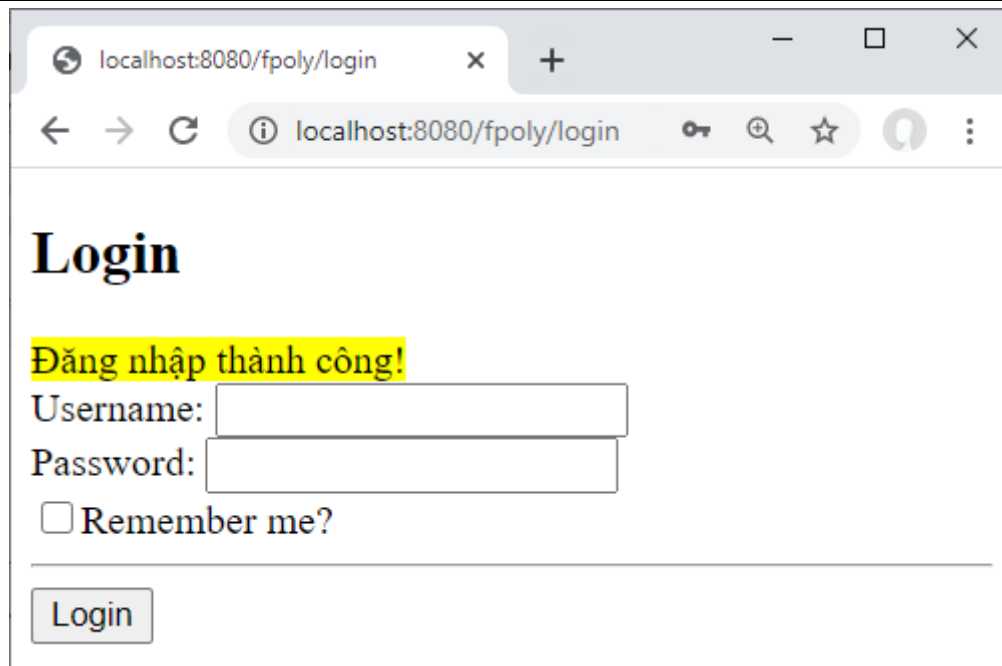
### BÀI 4: XỬ LÝ COOKIE

Xây dựng trang web đăng nhập có ghi nhớ tài khoản bằng cookie như mô tả ở các hình sau:

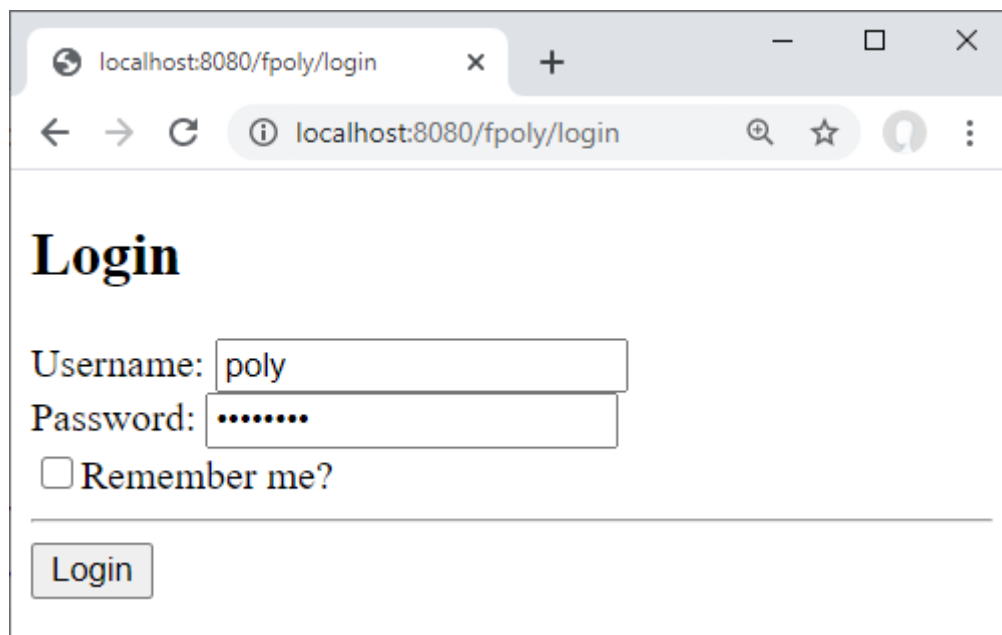


The screenshot shows a web browser window with the address bar displaying 'localhost:8080/fpoly/login'. The page content includes a 'Login' heading, followed by 'Username:' and 'Password:' labels next to input fields. Below these is a checkbox labeled 'Remember me?'. At the bottom of the form is a 'Login' button.

Hình 1: Form đăng nhập



Hình 2: Đăng nhập thành công



Hình 3: Hiển thị tài khoản đã ghi nhớ lên form

Hướng dẫn:

- Tạo LoginServlet.java, CookieUtils.java và login.jsp
- Tổ chức mã LoginServlet.java

```
@WebServlet("/login")
public class LoginServlet extends HttpServlet{
    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp)
        throws ServletException, IOException {
        // mã đọc cookie đã ghi nhớ ở đây
        req.getRequestDispatcher("/views/login.jsp").forward(req, resp);
    }

    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse resp)
        throws ServletException, IOException {
        // mã đăng nhập và ghi nhớ bằng cookie ở đây
        req.getRequestDispatcher("/views/login.jsp").forward(req, resp);
    }
}
```

- Thiết kế form login.jsp

```
<mark>${message}</mark>
<form action="/fpoly/login" method="post">
    Username: <input name="username"> <br>
    Password: <input name="password" type="password"> <br>
    <input name="remember" type="checkbox"> Remember me?
    <hr>
    <button>Login</button>
</form>
```

- Viết mã cho CookieUtils.java chứa các hàm tiện ích làm việc với Cookie

```
public class CookieUtils {
    // Tạo và gửi cookie về client để lưu
    public static Cookie add(String name, String value, int hours,
        HttpServletResponse resp) {
        Cookie cookie = new Cookie(name, value);
        cookie.setMaxAge(hours*60*60);
        cookie.setPath("/");
        resp.addCookie(cookie);
        return cookie;
    }
}
```

```
// Đọc giá trị cookie gửi từ client
public static String get(String name, HttpServletRequest req) {
    Cookie[] cookies = req.getCookies();
    if(cookies != null) {
        for(Cookie cookie: cookies) {
            if(cookie.getName().equalsIgnoreCase(name)) {
                return cookie.getValue();
            }
        }
    }
    return "";
}
```

- Bổ sung mã vào doPost() của LoginServlet.java

```
// đọc tham số form đăng nhập
String username = req.getParameter("username");
String password = req.getParameter("password");
String remember = req.getParameter("remember");
// kiểm tra tài khoản đăng nhập
if(!username.equalsIgnoreCase("poly")) {
    req.setAttribute("message", "Sai tên đăng nhập!");
}
else if(password.length() < 6) {
    req.setAttribute("message", "Sai mật khẩu!");
}
else {
    req.setAttribute("message", "Đăng nhập thành công!");
    // ghi nhớ hoặc xóa tài khoản đã ghi nhớ bằng cookie
    int hours = (remember == null) ? 0 : 30*24; // 0 = xóa
    CookieUtils.add("username", username, hours, resp);
    CookieUtils.add("password", password, hours, resp);
}
```

- Bổ sung mã vào doGet() của LoginServlet.java để đọc cookie và gửi sang login.jsp hiển thị sẵn

```
// đọc giá trị của cookie
String username = CookieUtils.get("username", req);
String password = CookieUtils.get("password", req);
```

*// chuyển sang login.jsp để hiển thị lên form*

```
req.setAttribute("username", username);  
req.setAttribute("password", password);
```

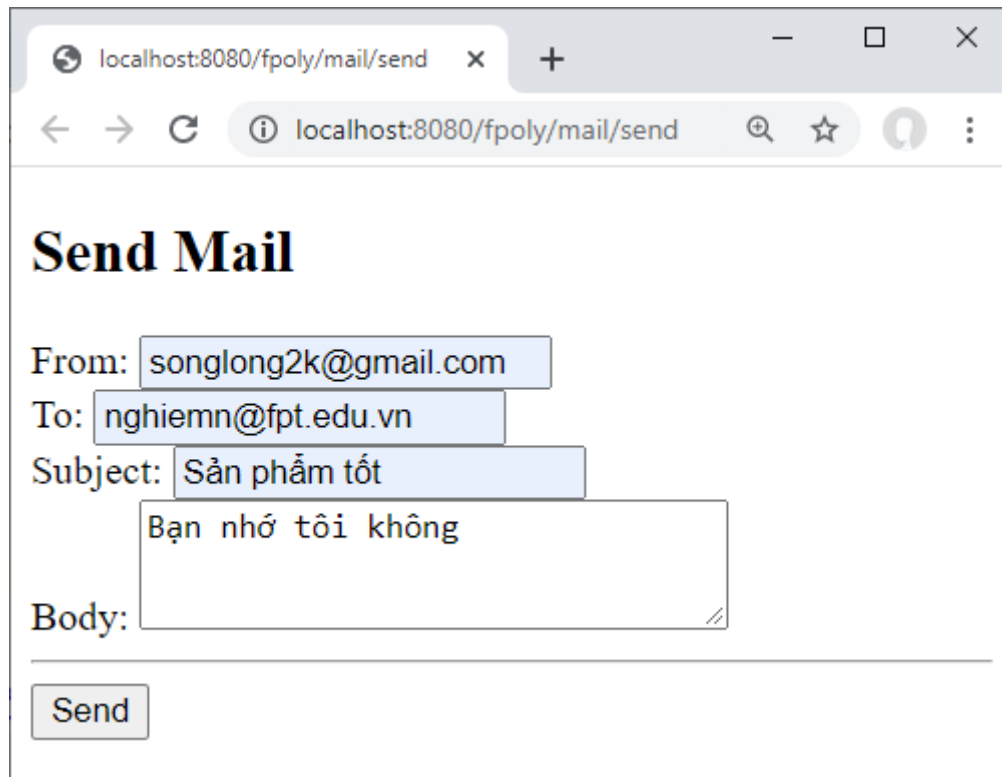
- Hiệu chỉnh form login.jsp để hiển thị cookie

Username: <input name="username" value="{username}"> <br>

Password: <input name="password" type="password" value="{password}"> <br>

## BÀI 5: GỬI EMAIL

Xây dựng trang web cho phép gửi email như mô tả ở các hình sau

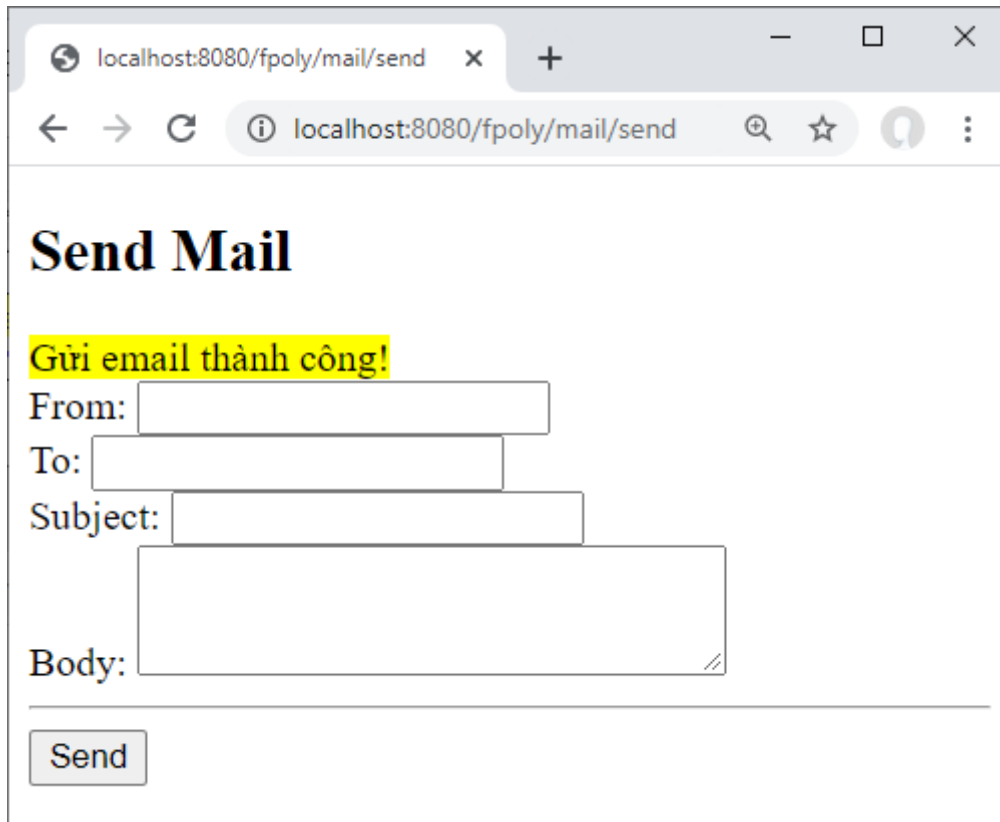


The screenshot shows a web browser window with the address bar displaying 'localhost:8080/fpoly/mail/send'. The page title is 'Send Mail'. The form contains the following fields:

- From: songlong2k@gmail.com
- To: nghiemn@fpt.edu.vn
- Subject: Sản phẩm tốt
- Body: Bạn nhớ tôi không

A 'Send' button is located at the bottom of the form.

Hình 1: Form gửi mail



localhost:8080/fpoly/mail/send

## Send Mail

**Gửi email thành công!**

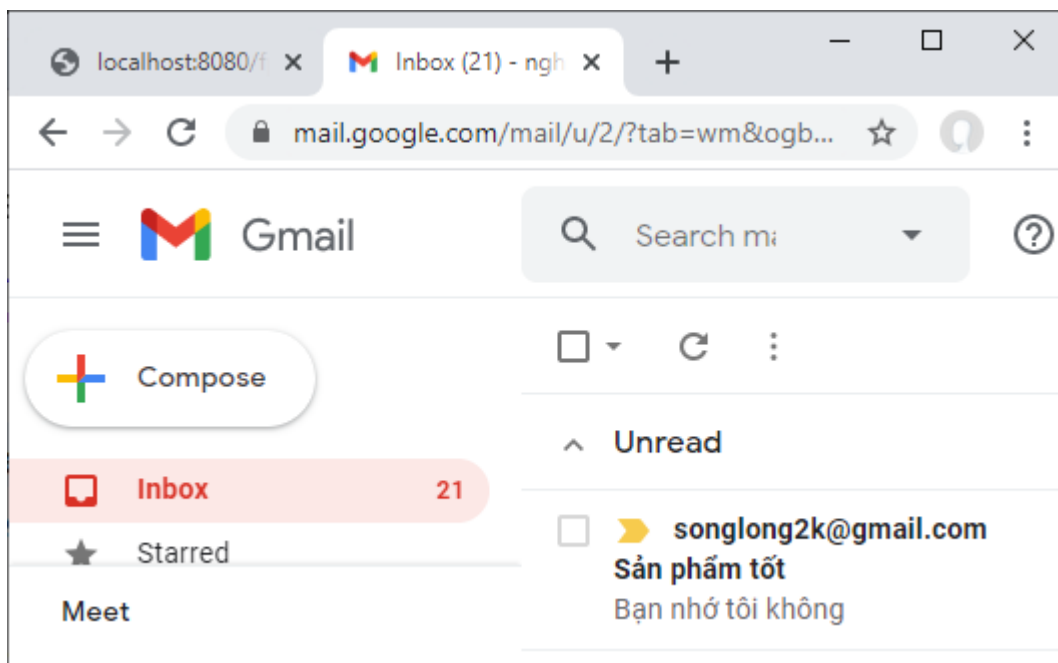
From:

To:

Subject:

Body:

Hình 2: Thông báo gửi thành công



Hình 3: Kết quả check mail