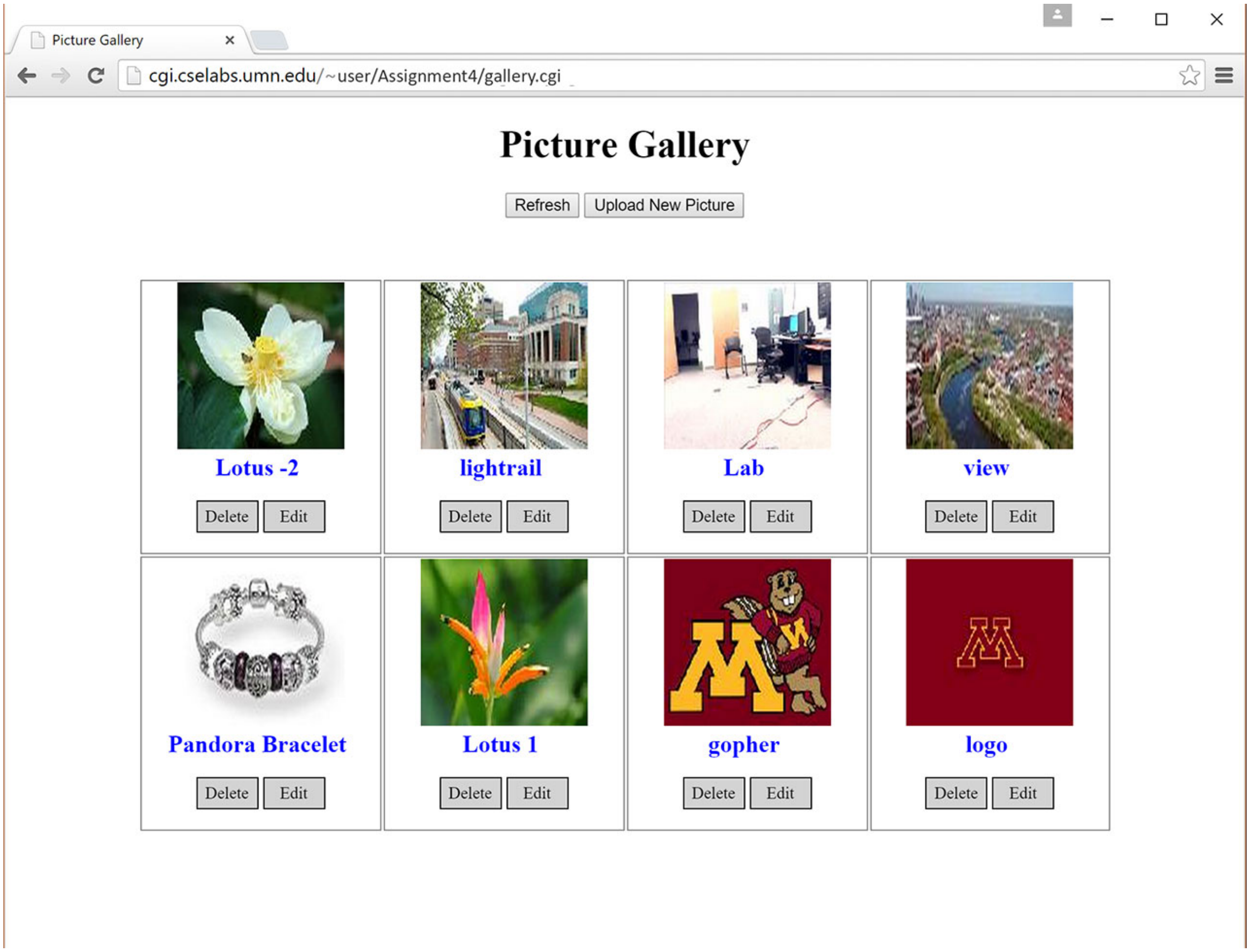
CSci-4131: Internet Programming, Fall 2015

Assignment 4: Programming with CGI

Due Date: October 28, 2015

This assignment is to be done __individually.__



Screenshot 1

Objective:

In this assignment you will learn about CGI programming using Python for generating HTML documents dynamically. You will also learn how to use JavaScript for form checking before submission.

Problem Statement:

You will create a website for uploading and viewing photos. It will support functionality for adding, deleting, and viewing photos. Your program will be hosted on a web server to provide photo upload and viewing functionalities. A user will be able to upload a number of photos and then view them individually. User will be able to upload new photos or delete some existing ones.

The server side functionalities will be implemented using Python. The client-side functionality will be implemented using JavaScript and you should be able to use a substantial part of your Assignment 2 work.

One major difference in this assignment from previous one, is the fact that you are not opening a new browser tab/window in any part of your different pages. Navigation between pages will be on the same tab, and for this to work on CGI scripts, you need to use the **server-side redirection** technique. HTTP protocol has a header command to redirect the user from one page to another called "Location."

Functionality:

On the server**,** your Python program will maintain a single folder called "pictures" where all the photos will be saved. It will also maintain files containing brief description (photo title in this case) for each photograph.

You are required to support the following functionality.

Gallery: **page will be dynamically generated by your Python/CGI program: gallery.cgi. Gallery will be the first page to be displayed by your program. (*Screenshot 1*)**

It would **display thumbnails** of all the photos in the "pictures" folder, along with the **photo display functionality**, similar to that you implemented in **Assignment 2**, but this time we only show a title line.

- Display previously created 140x140 pixels **thumbnails of photos** in a centered grid with 4 equal width columns.
- Clicking on the thumbnail should open a new DIV popup with JavaScript inside the Gallery page and **not** on a new tab. This DIV covers the whole page and shows the actual full-sized photo. To close the DIV, user just needs to click anywhere on the DIV, picture, or title. Check out **Screenshot 2** for more visual details at end of this write-up.
- Buttons to **edit title and delete photo** underneath each thumbnail.
- Photo **delete button** that links to delete.cgi page.
- Photo **edit button** enables editing the title of one individual photo at a time by clicking on the 'Edit' button. Clicking edit button should send the user to edit.cgi page.
- A button to "**Refresh View.**" This button doesn't do much if only one user is using your website, but if a few people are uploading photos, then this can refresh the table grid.
- An "**Upload button"** Clicking it would go to upload page which provides an interface for uploading a photo. This interface is described below.

Upload Page: This would present a form for photo upload. This can be either a static HTML document or a Python CGI script. (Screenshot 3)

- Your program will present a webpage to the user to upload a Photo file, it will present a form with the file upload function.

- This form should have the following elements:
- Input element of **“text” type** for photo title (100 characters,)
- HTML **file upload element** to upload a photo file. This will provide file browsing menu.
- You should provide one **upload button** to submit the selected photo file.
- You should provide one **cancel button** to cancel the upload and return to the Gallery.
- On “submit”, on the client-side your code should check whether the file being uploaded is indeed an image. You should use JavaScript to do this checking based on the selected file name.
- On the server-side, you should validate the title. If the title is empty, you should show the proper error message and reject the file upload (**Screenshot 4)**.
- On the server-side, after validation and when the file is saved, you need to also store the title of the photo and create a 140x140 pixels thumbnail of the full-sized picture. The thumbnail will be used in the ‘gallery.cgi’ page. The thumbnails can be created using Python “Image” library.
- **For simplifications, you can limit the accepted file type to JPEG photos.**

Edit page: This would present a form to edit the title of the image for which ‘Edit’ button is clicked. (Screenshot 5)

- Display the **current title** of the image in input “text” element when the page loads.
- A form that allows to modify the title of the image.
- 2 buttons – Edit and Cancel on the page.
- Title should be validated before saving in the image text file (**Screenshot 6)**
- After successful updating of the picture title, user needs to be redirected back to the gallery.cgi page via a **server-side redirect routine**.
- Clicking Cancel also redirects the user back to gallery.cgi page.

Delete: This would ask for confirmation to delete the image for which ‘Delete’ button is clicked. (Screenshot 7)

- Display the title of the photo that is being deleted, and a confirmation message.
- 2 options – Delete and Cancel
- After clicking Delete, all three files associated with each picture needs to be deleted from “pictures” folder: thumbnail, text file that stores the title, and full-sized image. After all the files are deleted, user needs to be redirected back to the gallery.cgi page via a **server-side redirect routine**.
- Clicking cancel also redirects the user back to gallery.cgi

Please take Note that all the buttons (edit, upload, delete) should open a new page/div in the same browser window/tab as the Gallery.

Design Guidelines:

- Please name the directory where you will upload the photos as “pictures”. You should create this directory in the current working directory on the server, where your Python/CGI program is stored.
- Use JavaScript to ensure that only image files are selected for upload. For simplifications, only accept JPEG files and reject all other file types.
- On the server side you will be using plain text files to store photo title.
- To simplify your solution, you can store photo, thumbnail, and title files using a filename based on the current time and date (hint: time.time() function). These files therefore will have the same date-time string, but different extensions, for example 1444438881.15.jpg, 1444438881.15_tn.jpg, and 1444438881.15.txt will represent full-sized picture, its 140x140 pixels thumbnail, and text file that has title of the photo.

Resources:

- Use Python’s datetime module to get the date-time: time.time() will return a number representing the current timestamp of the system. You can get string representation using str(time.time()) method.
- Use Python’s os module to list the contents of directory using listdir.

General Instructions:

- For testing your CGI program on the CSELab server, you will store it in a directory under your ‘.www’ directory. Make sure to protect this directory with 711 permissions.
- Although Python CGI extensions are usually “.py” CSELabs requires you to name all CGI scripts “.cgi” otherwise your programs won’t work.
- You upload your CGI scripts to the same CSELabs personal web folder, but you should access your files with a new address prefix: http://cgi.cselabs.umn.edu/~myuserXXX/
- So for example if you have put your gallery.cgi inside your “.www/4131/assignment4/” folder, then you should access it via: http://cgi.cselabs.umn.edu/~myuserXXX/4131/assignment4/gallery.cgi
- The log files are in /class/cgi/logs. You can look at the most recently written erro.log file, and do "tail -f error.log" to watch the error messages.

Grading Criteria:

- Photo view functionality:
  - 10 points for displaying 140x140 pixels thumbnails
  - 10 points for displaying full-sized photo with title in a DIV popup
  - 10 points for delete functionality and server-side redirection
  - 10 points for edit functionality, title validation, and server-side redirection
  - 5 points for refresh functionality
- Photo Upload
  - 20 points for uploading a photo file and validating the title on the server-side
  - 10 points for successfully creating a 140x140 pixels small thumbnail of the uploaded photo
  - 10 points for checking the photo file type on the client-side before uploading
  - 10 points for checking and validation photo “title”
- 5 points Code readability and structure

Submission Instructions:

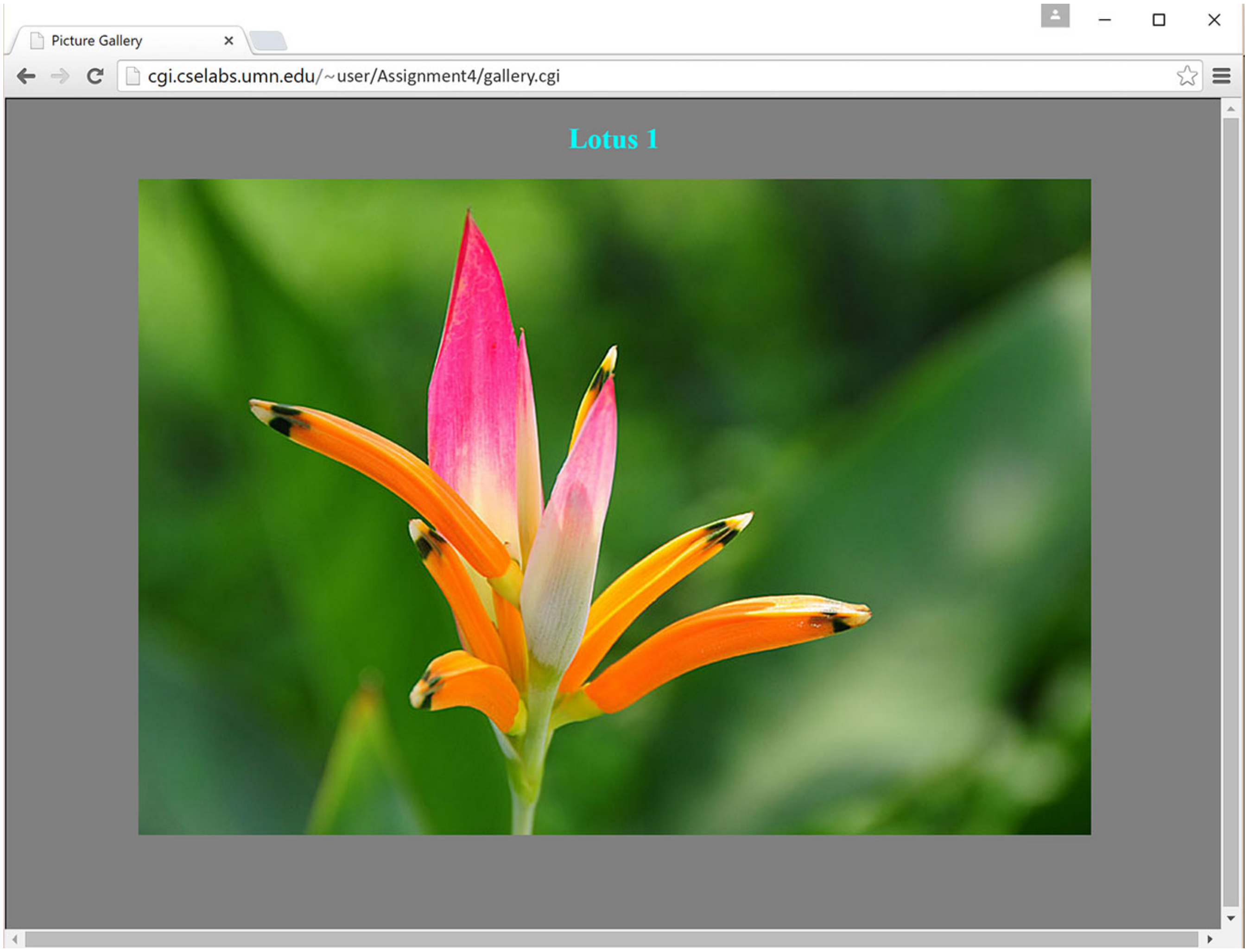
Include the following 6 files in your submission:

1. gallery.cgi
2. upload.cgi (or static upload.html plus upload.cgi handler)
3. edit.cgi
4. delete.cgi
5. styles.css
6. README file (optional)

In the README indicate the following things:

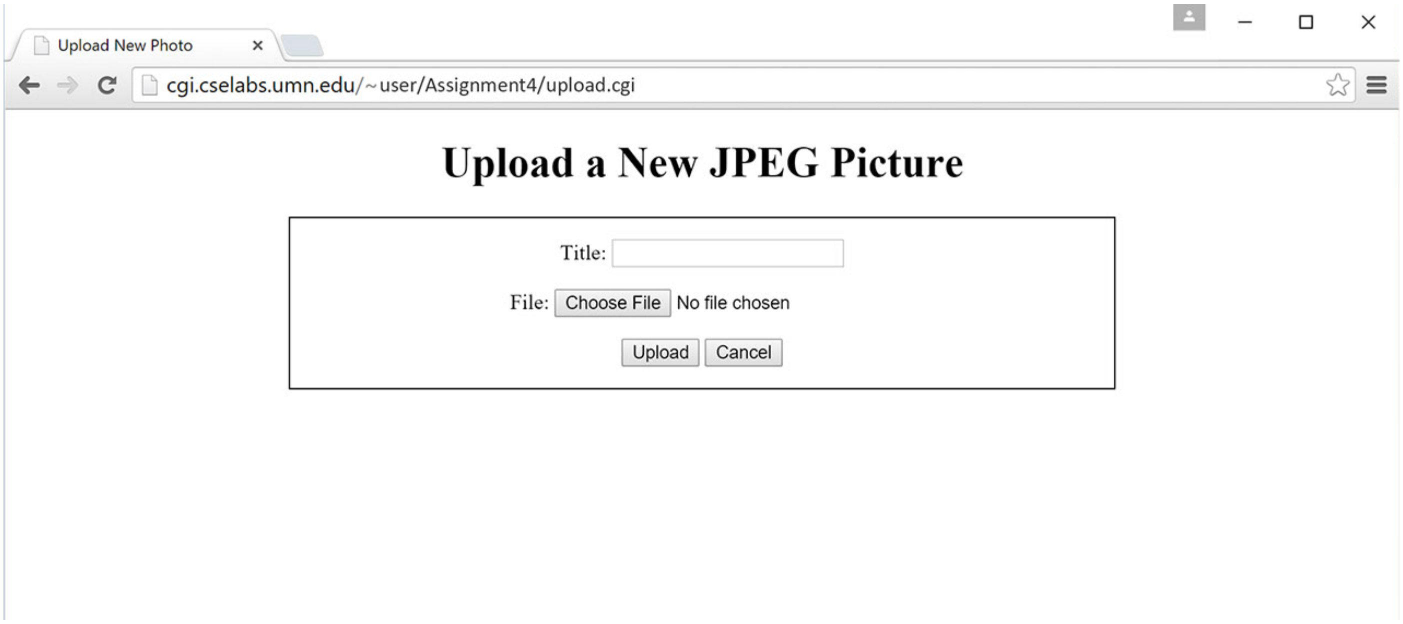
- Your Student Name and ID
- Any special instructions.

Submit the tarred directory containing all these files.



Lotus 1

Screenshot 2

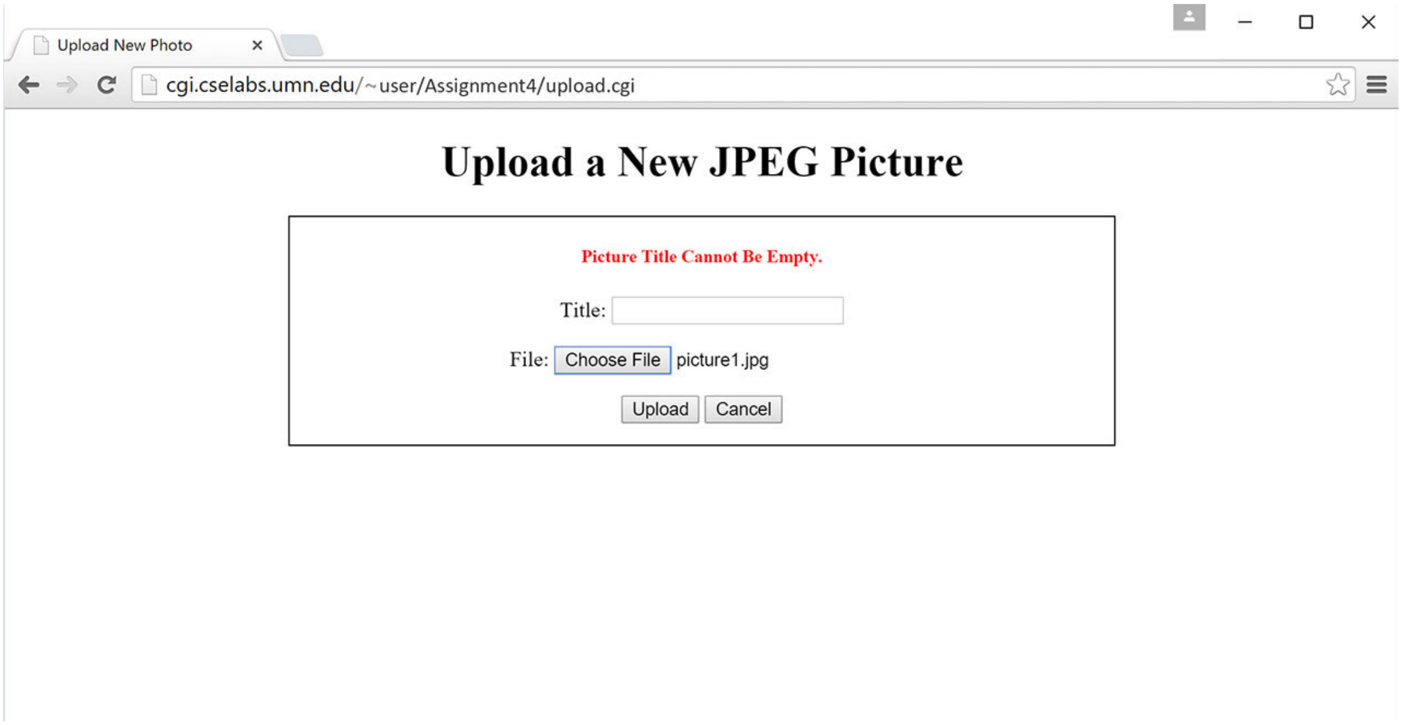


Upload a New JPEG Picture

Title:

File: No file chosen

Screenshot 3



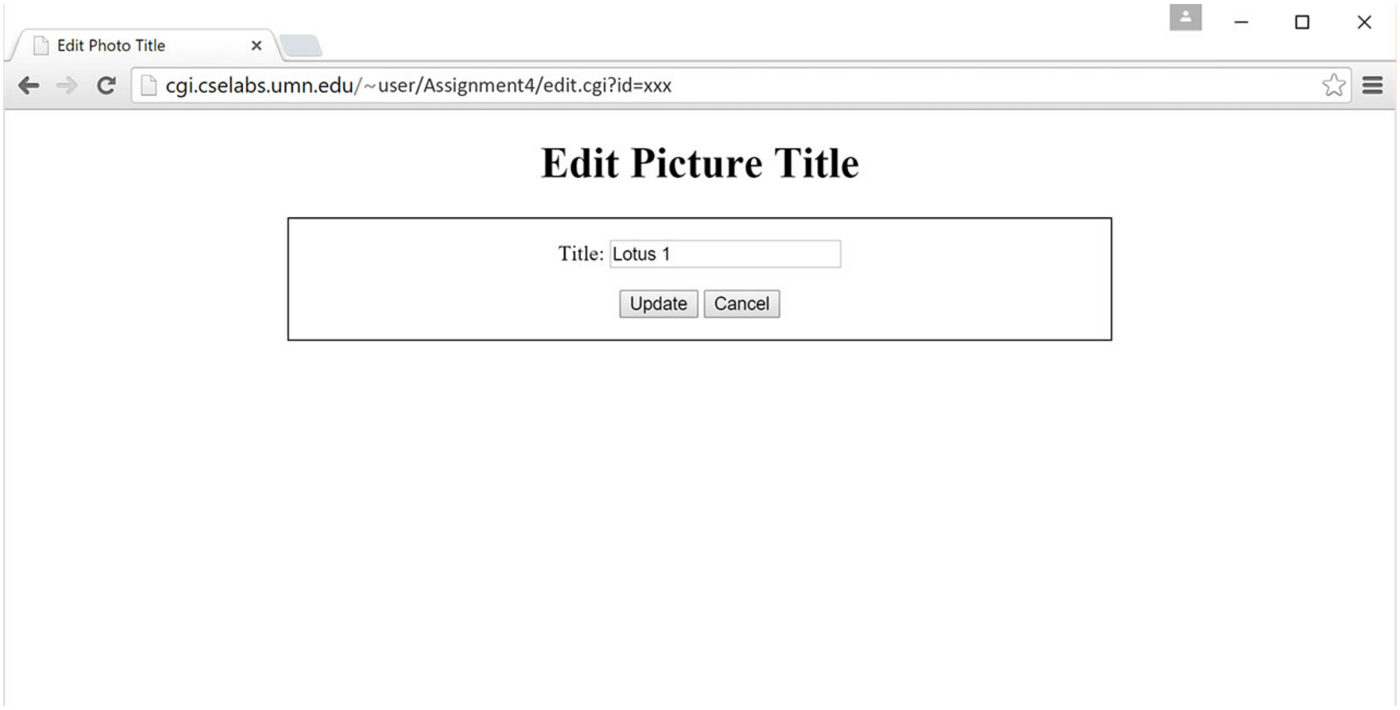
Upload a New JPEG Picture

Picture Title Cannot Be Empty.

Title:

File: picture1.jpg

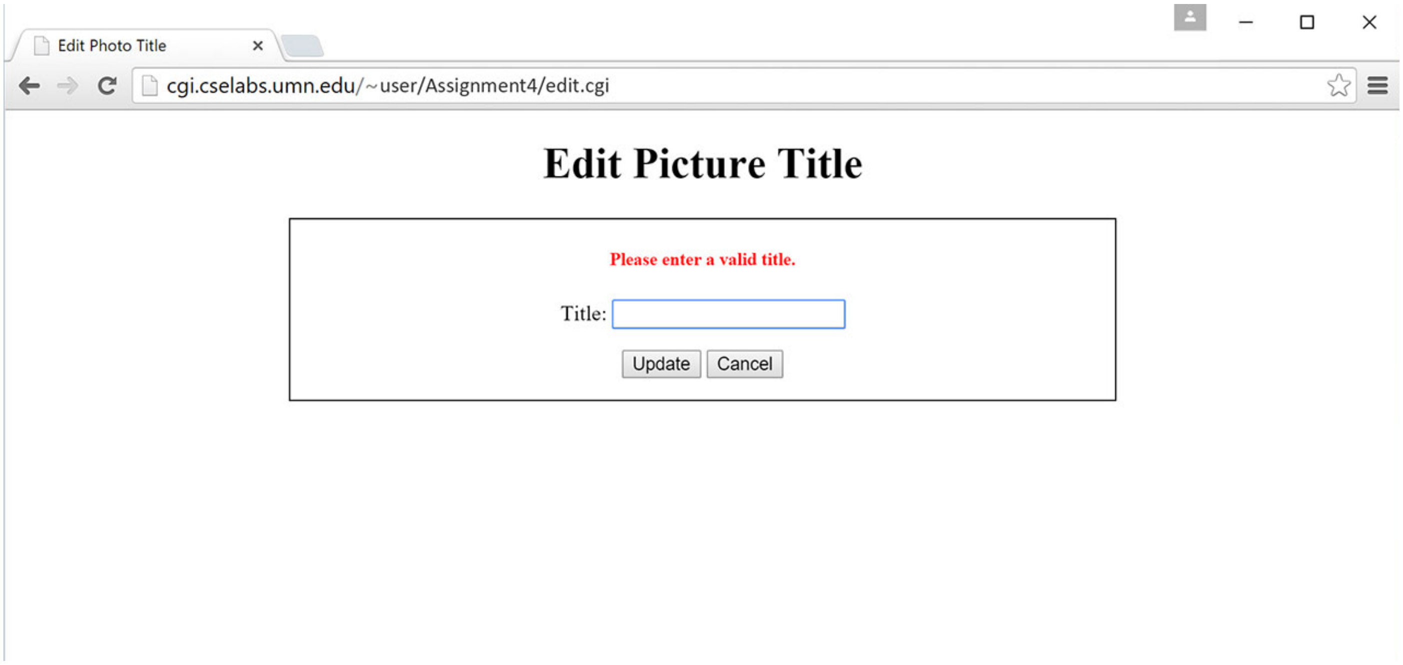
Screenshot 4



Edit Picture Title

Title: Lotus 1

Screenshot 5

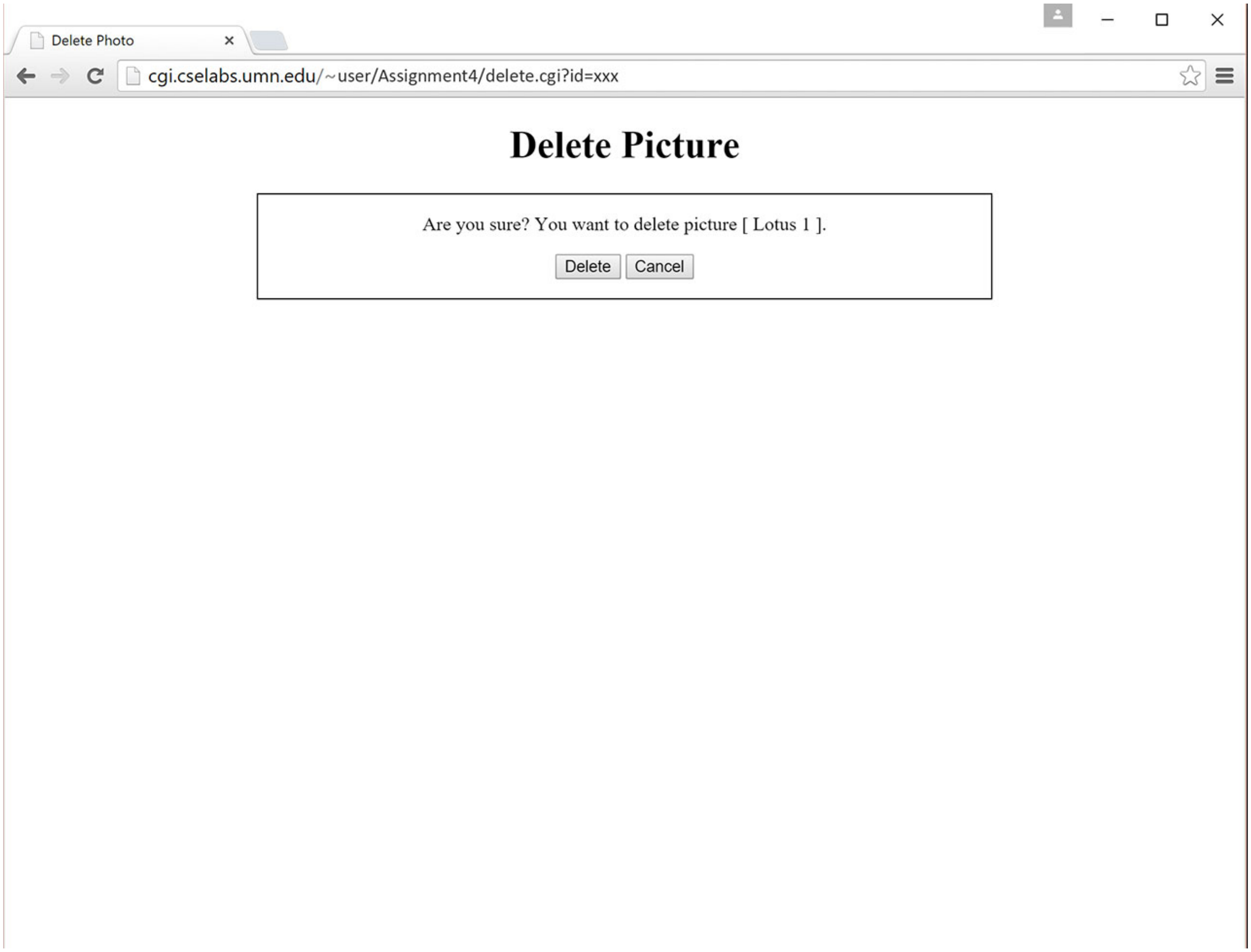


Edit Picture Title

Please enter a valid title.

Title:

Screenshot 6



Delete Picture

Are you sure? You want to delete picture [Lotus 1].

Screenshot 7