Project Tutorial 6
Mail and scheduled events

Dr. Yuzuko Nakamura

Dept. of Computer Science

Y.Nakamura@ucl.ac.uk

Mail functions

Sending mail with PHP

- In order to use PHP's mail functions, need an SMTP server.

- Two possibilities:
  - Install, configure, and run a mail server on localhost.
    - More difficult / time-consuming.
  - Connect your installation of PHP to an SMTP server someone else is running.

- Both of these may be hard to do with WampServer on lab machines.

PHP mail()

- In sendEmail.php:

```php
<?php
  $name = "Yuzuko Nakamura"; //sender's name
  $email = "sendersemail@gmail.com"; //sender's e-mail
address
  $recipient = "user@domain.name"; //recipient
  $mail_body= "Your password is $password"; //mail body
  $subject = "Lost password"; //subject
  $header = "From: ". $name . " <" . $email . ">\r\n";
//optional headerfields
  mail($recipient, $subject, $mail_body, $header); //mail
function
?>
```

Effect of mail() function

- Mail() prepares an email message and sends it to the mail server queue to send.
- It does not actually handle the sending of the email.
  - Error message from mail() will not be very informative.
  - Need to check the logs of your email server.
- Other mail libraries exist with more features (PEAR package, PHPMailer, etc.).
  - Specify authentication credentials, construct HTML emails more easily.

Sendmail program

- Simple mail-sending utility that comes with Linux.
  - Also bundled with XAMPP.
- For Windows, need to manually install something similar e.g. https://www.glob.com.au/sendmail/

Steps for connecting to a SMTP server

1. Edit sendmail.ini to add SMTP server name and port and authentication information (email and password).
2. Edit php.ini to add SMTP server details and sendmail binary location (Ctrl+F "mail function").
   – Restart localhost server after these changes.
3. Set up permissions on the sender's email to allow remote send requests.
4. Run PHP mail() code.

Example: using Google's mail servers

- Need to enable IMAP and also allow less secure app access.
- Information about Gmail SMTP server:

Outgoing Mail (SMTP) Server	smtp.gmail.com
	Requires SSL: Yes
	Requires TLS: Yes (if available)
	Requires Authentication: Yes
	Port for SSL: 465
	Port for TLS/STARTTLS: 587

Checking sendmail log

- On Linux systems, usually somewhere like /var/log/maillog or /var/log/mail.log
- For Windows sendmail, look for error.log in the sendmail folder.

Alternative to mail

- If you can't get mail working in your project, other possibilities:
  - Within-site notifications. Can be transient notifications or durable ones (stored somewhere in database).
  - Construct email but don't send it.

Timers and scheduled events

Executing a PHP script at regular intervals

- Multiple possible strategies:
  - Making a 'cron' job.
  - Using a MySQL database to schedule an event.
  - Writing a script in PHP (or some other language) that sleeps and wakes up and does work and goes back to sleep.

Cron jobs

- Linux-based systems come bundled with cron, a utility for automatically running commands at a specified frequency.

- The equivalent in Windows is the Task Scheduler.

- Your computer must be running for the job to run.

Making a new cron job

- From the terminal, edit the 'crontab' (cron table) file: `crontab -e`
- Add a cron job as a new line. Cron job format:
`*/5 * * * * /usr/bin/php ~/Sites/scripts/check_auctions.php`
  - First five arguments: what times to run this command (* means all): "every five minutes, every hour, every day, every month, every day of week."
  - Next arguments are the command and any arguments.
- Save file and quit.

Using Task Scheduler

- Has a graphical interface, fairly intuitive.
- The most frequently you can run a task is daily.

Running PHP files

- Need to find where the php binary/executable is on your computer (e.g. in *AMP directory, /bin).

- Or, instead of using php binary, can use wget or curl to request the page from your server, which will run it.

MySQL event scheduler

- Example:
  - ```
    SET GLOBAL event_scheduler = ON;
    CREATE EVENT event_name
       ON SCHEDULE
          EVERY 1 DAY
          STARTS '2020-12-31'
       DO
          [SQL statement];
    ```
- Multiple ways of specifying start/end times and intervals.
- Multiple SQL statements can be specified with some extra work.

Alternative to event scheduling

- Create a PHP script that does the appropriate work for updating an auction when it ends and manually run it.

Other checks

- Even if your auction ending script doesn't run very frequently, can use checks in PHP and SQL queries to:
  - Inform users loading the page that an auction has ended.
  - Prevent new bids on an auction from being added to the database after that auction ends.
- Fancier things can be done with JavaScript, but you don't need to bother for this project!

TODOs this week:

- Keep going with coursework requirements.
- Ask questions and get help when needed.