Visitor pages

For Hosts,
- Create a new schedule
- Update the schedule

For faculty members,
- View schedules for visitors (for 2006-07)
- View previous schedules for visitors (before 2006)
- Sign and update time slots

Created by Webmaster on 01/21/04.
Visitor pages

☞ The pages are “protected” by .htaccess with
shared username and password

disp
☞ the activities that a hosting faculty member might
need and what other faculty members’ might want to
do are distinguished
Hosting Faculty Member: Creating a Schedule

☞ A hosting faculty member can create a new schedule here
Hosting Faculty Member: Creating a Schedule

☞ This is a Perl script create.pl with a trivial C wrapper create.cgi

☞ It backends to a MySQL database called alumnibook
Hosting Faculty Member: Creating a Schedule

☞ The details for the visit can be updated here, including “opening up” time slots for other faculty members to sign up to meet with the visitor.
Hosting Faculty Member: Creating a Schedule

When the **SUBMIT** is pressed, the script `create.cgi` calls itself again with a parameter `reqType`
Hosting Faculty Member: Updating a Schedule

A hosting faculty member can update an existing schedule here.
Hosting Faculty Member: Updating a Schedule

☞ This is a Perl script `updatesch.pl` with a trivial C wrapper `updatesch.cgi`

☞ It also backends to the database `alumnibook`
Hosting Faculty Member: Updating a Given Schedule

☞ All details for the visit can be updated here, including “opening up” time-slots for meetings
Automatic Locking of a Given Schedule

If two people are editing a schedule simultaneously, this should be detected and an error message displayed to the second editor.
Automatic Locking of a Given Schedule

However this is currently fragile: it uses a single lockfile named dummylock
Faculty Recruitment

Divided into three sections:

☞ Applicant information
☞ References uploading
☞ Departmental activities
Faculty Recruitment: Applicant Information

Apply

Update an existing application
Faculty Recruitment: Creating an Application

☞ A Perl script `apply0220.pl` is called by a C wrapper `2020.cgi`.
Faculty Recruitment: Creating an Application

☞ The first time through, it displays create password message

☞ The same database alumnibook is used for both faculty recruitment and visitors
Faculty Recruitment: Creating an Application

☞ The first page of data entry for the applicant; most fields must be filled in
Faculty Recruitment: Creating an Application

☞ The filename for all of the files uploaded to the system is the one provided by the user, which we will have to change.
Faculty Recruitment: Creating an Application

☞ The fields on the second page are largely optional
Faculty Recruitment: Creating an Application

☞ The second application page is created by the same script **apply0220.pl**
Faculty Recruitment: Creating an Application

Likewise, the third application page is created by the same script `apply0220.pl`
Faculty Recruitment: Creating an Application

☞ The final page is also created by the same script apply0220.pl
Faculty Recruitment: Updating an Application

 Updating is done via Perl script `updateinfo0220.pl`, which has a trivial C wrapper `updateinfo2020.cgi`
Faculty Recruitment: Updating an Application

The user logs in with the same information as previously supplied.
Faculty Recruitment: Updating an Application

☞ The Perl script `updatein0220.pl` creates all of the various pages for updating
Faculty Recruitment: Updating an Application

☞ The user will be warned if files are missing
Faculty Recruitment: Updating an Application
Faculty Recruitment: Reviewing applications

Department members can read and comment on applications
Faculty Recruitment: Reviewing applications

☞ The Perl script `viewnew.pl` is wrapped by `viewnew.cgi`
Faculty Recruitment: Reviewing applications

_department members can use many criteria to select applications for review_
Faculty Recruitment: Reviewing applications
Faculty Recruitment: Reviewing applications

Faculty members can make comments about each application.
If the status is updated to anything other than “Reject”, “Rejected”, “ToBeRejected”, or “OnHold”, then a request for references is sent out.
Summary

What do you need to run these systems?
- Linux server
- Perl with CGI library
- Apache
- MySQL server

The code is Computer Science-specific (especially such items as lists of areas of interest) and will need to be customized for other departments.
The code is not designed for large numbers of users. Issues such as filenames, locking, and even table key fields need careful scrutiny.