Case Study: MyUniv – Training Mgmt System

Group 4

Team member Emp ID	**Team member Name**	**Team Member Email Id**
46292706	**Kishandeep Swain**	**kishandeep.swain@capgemini.com**
46289314	Priyanka Priyadarshini	priyanka.a.priyadarshini@capgemini.com
46283127	Sipra Subhadarsini Biswal	sipra-subhadarsini.biswal@capgemini.com
46289246	Sonali Dalai	sonali.dalai@capgemini.com
46284372	Purnashree Parikhita Pani	purnashree-parikhita.pani@capgemini.com
46283110	Lipsarani Sutar	lipsarani.sutar@capgemini.com

PROBLEM STATEMENT

1.2 OBJECTIVE

To create an online training management system like Udemy where employees can login and learn from training programs. The application is to be develope as executable file compiled on Linux.. There are 2 entities Admin and User.

1.3 ABSTRACT OF THE PROJECT

1. Admin should be able to create and update courses.
2. Admin should be able to view statistics for how many courses enrolled, most taken course, etc.
3. User should be able to login and browse through catalog.
4. Provide search filters and sorting for courses.
5. Details about the course and chapters should be provided to users.
6. User should be able to enroll for the selected course.
7. Track the progress of user like how many chapters completed, etc.
8. Allow user to add a bookmark to a particular chapter.
9. Handle data and errors properly. Show appropriate messages to user.
10. Display good input, output messages and reports in proper format.
11. Security features should be implemented whereever possible. For example user passwords can be stored in encrytped format.

1.4 FUNCTIONAL COMPONENTS OF THE PROJECT

Following is a list of functionalities of the system. Wherever, the description of functionality is not adequate; you can make appropriate assumptions and proceed.

When MyUniv starts it displays Following Screen -

```
-------------------------Login Screen--------------------------
1. Register new user
2. Login
0. Quit
```

System should maintain comma separated users.txt where each line stores username, password and all other information about registered user. Be default this file contents one line – admin,admin,..... – denoting admin user.

Whenever username and password is entered, system authenticates it with entry in “users.txt” file.

- If match is found and if admin user then “Admin Screen” is displayed.
- If match is found and if not admin user then “User Screen” is displayed.
- It match is not found then message “Invalid User or password” is displayed and system exits.

2. When admin user login MyUniv starts it displays “Admin Screen”

```
 ------------------------Admin Screen-------------------------
     1. Manage courses
     2. Manage chapters
     3. Link chapters to courses
     4. Display course statistics
     5. Display users statistics
     0. Quit

     Enter your option : <option>
```

option = 1 (Manage courses)

This further displays sub-menu -

```
    ------------------------Courses Screen-------------------------
     1. Add New Course
     2. Modify Course
     3. Delete Course
     0. Quit
```

Write all information regarding courses in “courses.txt” file where each line corresponds to one course. Information about courses to be stored as comma separated fields.

option = 2 (Manage chapters)

This further displays sub-menu -

```
    ------------------------Chapter Screen-------------------------
     1. Add New chapter
     2. Modify chapter
     3. Delete chapter
     0. Quit
```

Write all information regarding chapters in “chapters.txt” file where each line corresponds to one course. Information about chapters to be stored as comma separated fields. Filename along with pathname of training files (pdf/ video/ ppt etc..) for the chapter should be stored in chapters.txt file along with other information.

Option = 3 (Link chapters to courses)

This further displays sub-menu -

```
-------------------------Link Screen--------------------------
1. Add new link
2. Modify link
3. Delete link
0. Quit
```

option = 1 (Add new link)
Enter course number <courseNo>
Enter chapter number <chapterNo>
Enter serial number of the chapter in course <serialNo>

Write all information regarding chapters in “contents.txt” file where each line corresponds to course and chapter link. Information about chapters to be stored as comma separated fields.

For Example -
If course C0001 requires chapters intro, unixintro, awk, processintro in that order then in “contents.txt”

```
C0001,intro,1
C0001,unixintro,2
C0001,awk,3
C0001,processintro,4
```

option = 2 (Modify link)
Enter course number <courseNo>
Enter chapter number <chapterNo>
Enter new serial number of the chapter in course <serialNo>

Will update corresponding line for course number and chapter number with new serialNo in “contents.txt”.

option = 3 (Delete link)
Enter course number <courseNo>
Enter chapter number <chapterNo>

Will delete corresponding line for course number and chapter number from “contents.txt”

2. When user login MyUniv starts it displays “User Screen”

```
-------------------------User Screen--------------------------
1. Search course
2. Enroll course
3. Mark completed chapter
4. Add Bookmark
5. Display completed courses
```

0. Quit
Enter your option : <option>

option = 1 (Search course)

At least 10 filters should be provided in sub-menu to search courses from "courses.txt" file and corresponding chapters from "contents.txt" and "chapters.txt" files.

option = 2 (Enroll course)
Enter course name - <CourseName>

Write all information regarding enrollment in "enrollments.txt" file where each line corresponds to one enrollment. Information like username, <courseName>, date of enrollment etc to be stored as comma separated fields.

option = 3 (Mark completed chapter)
Enter course name - <CourseName>
Enter chapter name - <ChapterName>

Write all information regarding chapter completion in "completion.txt" file where each line corresponds to one completion. Information like username, <courseName>, <ChapterName>, date of completion etc to be stored as comma separated fields.

If all chapters are completed then course is marked as "completed" in "enrollments.txt".

option = 4 (Add Bookmark)
Enter course name - <CourseName>
Enter chapter name - <ChapterName>
Bookmark page number - <PageNo>

Write all information regarding chapter completion in "bookmarks.txt" file where each line corresponds to one bookmark. Information like username, <courseName>, <ChapterName>, date of completion etc to be stored as comma separated fields.

option = 5 (Display completed courses)
Display courses completed by user from "enrollments.txt" file.

Assumptions: <Write assumptions made>

Technical Requirements -

1) C programming language
2) Use file input/output operations to read and write data.
3) Use multiple Linked Lists to read data from files at the beginning and write updated data to

files before applicaton ends.

4) Use dynamic memory allocation.

<u>Non Functional Requirements</u>

1) Multi-file multi-directory solution is expected. Modular and maintainable code (comments) and all coding standards should be followed.

2) makefile to build application. Two-step compilation process - .o and then executable should be generated.

3) Use valgrind tool on application executable to detect memory leak. Final valgrind report to be submitted in "reports" directory.

4) Level 0 DFD (context diagram), Level 1 DFD, Flow diagram and pseudocode for 2 complex functions logic.

5) SRS in pdf format, RTM, Plan, Presentation. MOMs

6) HLD_LLD Document (optional)

7) Design review, Code review, Inspection Logs of design and code reviews

8) Unit test cases and Integration test cases in UT_IT document. Both types of test cases i.e. sunny and rainy should be present in this document

Set Up Checklist for Project

Software Requirement:

Vi Editor, ctags, splint, valgrind, gcc, make, git account

Minimum System / Hardware Requirements:

Laptop with access to internet and Linux OS