3.4.4 Definition of tables in the database

```
CREATE TABLE IF NOT EXISTS `award`
(
  `id` varchar(10) NOT NULL,
  `award_id` int(11) NOT NULL,
  `award_name` varchar(80) NOT NULL,
  `award_agency` varchar(80) NOT NULL,
`url` varchar(40) DEFAULT NULL,
  `remark` varchar(100) DEFAULT NULL,
  `file` varchar(250) DEFAULT NULL,
  `award_date` date DEFAULT NULL,
  PRIMARY KEY (`id`,`award_id`),
  KEY `aid` (`award_id`)
) ENGINE=InnoDB DEFAULT
CHARSET=latin1;


CREATE TABLE IF NOT EXISTS `book` (
  `book_id` int(11) NOT NULL,
  `role` varchar(20) NOT NULL,
  `book_or_chapter` varchar(20) NOT
NULL,
  `book_title` varchar(20) NOT NULL,
  `book_edition` varchar(20) NOT NULL,
  `publisher_name` varchar(20) NOT NULL,
  `isbn` varchar(25) NOT NULL,
  `id` varchar(10) NOT NULL,
  `book_date` date DEFAULT NULL,
PRIMARY KEY (`book_id`,`id`),
  UNIQUE KEY `isbn` (`isbn`),
  KEY `id` (`id`)
) ENGINE=InnoDB DEFAULT
CHARSET=latin1;


CREATE TABLE IF NOT EXISTS
`book_authors` (
  `id` varchar(10) NOT NULL,
  `book_id` int(11) NOT NULL,
  `author_number` int(11) NOT NULL,
  `author_name` varchar(40) NOT NULL,
  PRIMARY KEY
(`id`,`book_id`,`author_number`),
  KEY `book_authors_ibfk_2` (`book_id`)
) ENGINE=InnoDB DEFAULT
CHARSET=latin1;


CREATE TABLE IF NOT EXISTS
`book_chapters` (
  `id` varchar(10) NOT NULL,
  `book_id` int(11) NOT NULL,
  `chapter_name` varchar(50) NOT NULL,
  PRIMARY KEY
(`book_id`,`id`,`chapter_name`),
  KEY `book_chapters_ibfk_2` (`book_id`),
  KEY `id` (`id`)
) ENGINE=InnoDB DEFAULT
CHARSET=latin1;


CREATE TABLE IF NOT EXISTS
`communityuser` (
  `community_user_id` int(11) NOT NULL,
  `name_of_event` varchar(40) NOT NULL,
  `role` varchar(40) DEFAULT NULL,
  `location` varchar(40) DEFAULT NULL,
  `date_of_event` date DEFAULT NULL,
  `url` varchar(40) DEFAULT NULL,
  `additional_information` varchar(40)
```

```
DEFAULT NULL,
  `id` varchar(10) NOT NULL,
  PRIMARY KEY
(`community_user_id`,`id`),
  KEY `id` (`id`)
) ENGINE=InnoDB DEFAULT
CHARSET=latin1;


CREATE TABLE IF NOT EXISTS
`conference` (
  `id` varchar(10) NOT NULL,
  `conference_id` int(11) NOT NULL,
`conference_type` varchar(100) NOT
NULL,
  `research_area` varchar(100) NOT NULL,
  `paper_associated_project` varchar(5) NOT
NULL,
  `project_name` varchar(100) DEFAULT
NULL,
  `conference_name` varchar(200) NOT
NULL,
  `organizer` varchar(100) DEFAULT
NULL,
  `from_date` date NOT NULL,
  `to_date` date NOT NULL,
  `venue` varchar(100) NOT NULL,
  `abstract` varchar(500) NOT NULL,
  `keywords` varchar(200) NOT NULL,
  `url` varchar(100) DEFAULT NULL,
  `file` varchar(500) NOT NULL,
  `from_page` int(11) DEFAULT NULL,
  `to_page` int(11) DEFAULT NULL,
  `paper_title` varchar(100) DEFAULT
NULL,
  PRIMARY KEY (`id`,`conference_id`),
  KEY `ctype` (`conference_type`),
  KEY `cid` (`conference_id`)
) ENGINE=InnoDB DEFAULT
CHARSET=latin1;


CREATE TABLE IF NOT EXISTS
`conference_authors` (
  `id` varchar(10) NOT NULL,
  `conference_id` int(11) NOT NULL,
  `author_name` varchar(40) NOT NULL,
  PRIMARY KEY
(`id`,`conference_id`,`author_name`),
  KEY `con_id` (`conference_id`)
) ENGINE=InnoDB DEFAULT
CHARSET=latin1;


CREATE TABLE IF NOT EXISTS
`consultancy` (
  `consultancy_id` int(11) NOT NULL,
  `client` varchar(80) NOT NULL,
  `work_title` varchar(80) NOT NULL,
  `start_date` date NOT NULL,
  `end_date` date NOT NULL,
  `revenue_generated` varchar(15) NOT
NULL,
  `summary` varchar(100) DEFAULT
NULL,
  `labs_allocated` varchar(100) DEFAULT
NULL,
  `url` varchar(40) DEFAULT NULL,
  `id` varchar(10) NOT NULL DEFAULT
",‘’  PRIMARY KEY (`consultancy_id`,`id`),
KEY `id` (`id`)
) ENGINE=InnoDB DEFAULT
CHARSET=latin1;
```

```
CREATE TABLE IF NOT EXISTS
`consultancy_collaboration` (
  `id` varchar(10) NOT NULL,
  `consultancy_id` int(11) NOT NULL,
  `collaborator_name` varchar(40) NOT
NULL,
  PRIMARY KEY (`id`,`consultancy_id`),
  KEY `consultancy_id` (`consultancy_id`)
) ENGINE=InnoDB DEFAULT
CHARSET=latin1;


CREATE TABLE IF NOT EXISTS
`consultancy_faculty_involved` (
  `id` varchar(10) NOT NULL,
  `consultancy_id` int(11) NOT NULL,
  `fname` varchar(40) NOT NULL,
  PRIMARY KEY
(`id`,`consultancy_id`,`fname`),
  KEY `consultancy_fac_inv_ibfk_3`
(`consultancy_id`)
) ENGINE=InnoDB DEFAULT
CHARSET=latin1;


CREATE TABLE IF NOT EXISTS
`consultancy_student_involved` (
  `id` varchar(10) NOT NULL,
  `consultancy_id` int(11) NOT NULL,
  `student_usn` varchar(10) NOT NULL,
  PRIMARY KEY
(`id`,`consultancy_id`,`student_usn`),
  KEY `consultancy_id` (`consultancy_id`)
) ENGINE=InnoDB DEFAULT
CHARSET=latin1;


CREATE TABLE IF NOT EXISTS
`courses_list` (
  `course_id` varchar(8) NOT NULL,
  `course_name` varchar(40) NOT NULL,
  `semester` int(11) NOT NULL,
  `ug/pg` varchar(2) NOT NULL,
  `syllabus_year` year(4) NOT NULL,
  PRIMARY KEY (`course_id`)
) ENGINE=InnoDB DEFAULT
CHARSET=latin1;


CREATE TABLE IF NOT EXISTS
`courses_taught` (
  `course_id` varchar(8) NOT NULL,
  `academic_year` year(4) DEFAULT
NULL,
  `number_of_students` int(4) DEFAULT
NULL,
  `pass_percent` decimal(5,2) NOT NULL,
  `id` varchar(10) NOT NULL DEFAULT
",‘’  PRIMARY KEY (`id`,`course_id`),
  KEY `course_id` (`course_id`)
) ENGINE=InnoDB DEFAULT
CHARSET=latin1;


CREATE TABLE IF NOT EXISTS
`faculty_exchange_program` (
  `id` varchar(10) NOT NULL,
  `institution` varchar(40) NOT NULL,
  `subject` varchar(20) NOT NULL,
  `start_date` date NOT NULL,
```

```
  `end_date` date NOT NULL,
  `ug_pg` varchar(20) NOT NULL,
  `collaboration_type` varchar(20) NOT
NULL,
  `details_of_collaboration` varchar(20)
DEFAULT NULL,
  `feedback` text NOT NULL,
  PRIMARY KEY (`id`) ) ENGINE=InnoDB
DEFAULT CHARSET=latin1;


CREATE TABLE IF NOT EXISTS
`faculty_member` (
  `fid` varchar(10) NOT NULL,
  `username` varchar(50) NOT NULL,
`password` varchar(60) NOT NULL,
;
  `name` varchar(30) NOT NULL,
  `address` varchar(100) NOT NULL,
  `phone_number` varchar(30) NOT NULL,
  `picture` varchar(100) NOT NULL,
`gender` varchar(100) NOT NULL,
  `email` varchar(30) NOT NULL,
  `date_of_join` date NOT NULL,
  `date_of_birth` date DEFAULT NULL,
  `department` varchar(3) DEFAULT NULL,
  PRIMARY KEY (`fid`,`username`),
  KEY `fid` (`fid`)
) ENGINE=InnoDB DEFAULT
CHARSET=latin1
```

3.4.5 Definition of relations and the cardinality ratios of the participating entities

Sl. No.	Participating Entity 1	Relation	Participating Entity 2	Cardinality Ratio
1.	Faculty	Has	Profile	1:1
2.	Faculty	Has won	Awards	M:N
3.	Faculty	Attended/Conducted	Conference	M:N
4.	Faculty	Part of/Conducted	Projects	M:N
5.	Faculty	Was Part of	Faculty_Exchange	M:N
6.	Faculty	Authored	Book/Journals	M:N
7.	Faculty	Teaches	Courses	M:N

Table 3.4 Cardinality Ratios

Chapter 4

IMPLEMENTATION

4.1 Client-Server Specification

The software runs on Client Server Architecture where the server side consists of php scripts running on the server. The server can be run on windows as well as linux based systems. It works best with Apache server 2.4, php 5.5.9 and Mysql server 5.5.xx installed on the system.

The clients can use any AJAX compatible browser. Further specifications for the browsers are listed below:

Mozilla Firefox – Version 31

Google Chrome – Version 44

Having the above mentioned versions of the browsers at the client side enables the use of AJAX and JQuery which is used for the development of the software.

The software developed has fat server architecture, which implies that the clients only need to have minimal configuration for their systems.

4.2 Technologies

The various technologies used for development purposes are:

4.2.1 HTML5

- HTML5 is a markup language used for structuring and presenting content on the World Wide Web. It was finalized, and published, on 28 October 2014 by the World Wide Web Consortium.
- Its core aims are to improve the language with support for the latest multimedia while keeping it easily readable by humans and consistently understood by computers and devices.
- Various new APIs have been included :- Geolocation, Local Storage, etc.

4.2.2 Javascript

- JavaScript is a lightweight, interpreted programming language. It is designed for creating network-centric applications.
- JavaScript is very easy to implement because it is integrated with HTML. It is open and cross-platform.

- Client-side JavaScript is the most common form of the language. The script should be included in or referenced by an HTML document for the code to be interpreted by the browser.
- JavaScript can be used to trap user-initiated events such as button clicks, link navigation, and other actions that the user initiates explicitly or implicitly.
- We can create interfaces that react when the user hovers over them with a mouse or activates them via the keyboard.

4.2.3 JQuery

- JQuery is a fast and concise JavaScript library created by John Resig in 2006. jQuery simplifies HTML document traversing, event handling, animating, and Ajax interactions for Rapid Web Development.
- The JQuery offers an elegant way to capture a wide variety of events, such as a user clicking on a link, without the need to clutter the HTML code itself with event handlers.
- The JQuery helps you a lot to develop a responsive and feature-rich site using AJAX technology.
- JQuery is very lightweight library - a–out 19KB in size.

4.2.4 AJAX

- Ajax is a set of web development techniques utilizing many web technologies used on the client-side to create asynchronous Web applications.
- With Ajax, web applications can send data to and retrieve from a server asynchronously (in the background) without interfering with the display and behavior of the existing page.
- Ajax is not a technology, but a group of technologies. HTML and CSS can be used in combination to mark up and style information. The DOM is accessed with JavaScript to dynamically display – and allow the user to interact with – the information presented.

4.2.5 XML

- XML stands for E**X**tensible **M**arkup **L**anguage.
- XML is designed to store and transport data.
- XML is designed to be self-descriptive.
- XML is designed to be both human- and machine-readable.
- XML is a W3C Recommendation.

4.2.6 CSS

- Cascading Style Sheets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable.
- CSS saves time.
- Pages load faster.
- Easy maintenance - T– make a global change, simply change the style, and all elements in all the web pages will be updated automatically.
- Offline Browsing - C–S can store web applications locally with the help of an offline catche.Using of this, we can view offline websites.The cache also ensures faster loading and better overall performance of the website.
- Platform Independence - T–e Script offer consistent platform independence and can support latest browsers as well.

4.2.7 JSON

- JSON or JavaScript Object Notation is a lightweight text-based open standard designed for human-readable data interchange.
- It is used while writing JavaScript based applications that includes browser extensions and websites.
- It is primarily used to transmit data between a server and web applications.
- Web services and APIs use JSON format to provide public data.
- It can be used with modern programming languages.

4.2.8 PHP

The programming language used for the development work is PHP. The reason for selection of this language includes among many others the following few.

- Open Source, PHP is completely free.
- PHP can be easily embedded directly into HTML.
- Platform independent can run on Windows Linux or Mac servers.
- Run faster on the internet and easily integrate AJAX, Callback etc.
- Interfaces very easily with Apache/MySQL
- Lots of good books and on-line help.
- It's available with documentation in many languages.
- Easy to learn compared to many other scripting languages. It has a syntax that is easy to parse and is actually rather human-friendly.

- Lots of hosting services have it ready to use, no special configuration.Pretty easy to access other web-based tools through PHP i.e. google maps, etc

4.2.9 MySQL

MySQLi is a relational data base management system (RDBMS) that runs as a server providing multi-user access to a number of databases.

- MySQL is an open source tool.
- MySQL is a popular choice of database for use in web applications, and is a central component of the widely-used LAMP web application software stack — LAMP is an acronym for "Linux, Apache, MySQL, PHP.
- MySQL is primarily an RDBMS and therefore ships with no GUI tools to administer MySQL databases or manage data contained within.
- Microsoft SQL server Express Management studio provides an ease in creating tables by a graphical as well as query based interface.

Chapter 5

TESTING

Software Testing is the process used to help identify the correctness, completeness, security and quality of the developed computer software. Testing is the process of technical investigation and includes the process of executing a program or application with the intent of finding errors.

Test Strategies

Test strategy tells the test plan of the project. It also tells how to test and what to test. The testing done in this project are Unit testing and Integration testing.

- Features to be tested: Form navigation and generation of reports.
- Items to be tested: Functioning of forms and buttons.
- Purpose of testing: To check the effective working of FMS.
- Pass / Fail Criteria: Changes made on the back end like recreation of tables should affect the front end as well. If so, the test is successful.
- Assumptions and Constraints: Tables should be created and values have to be entered at the back end before testing and entity integrity and referential integrity constraints should be taken care

5.1 Unit Testing

Unit testing is a software verification and validation method in which a programmer tests if individual units of source code are fit for use. Some of the tests performed in the project are inserting, delete, retrieve and modify.