

6.1 APPLICATION DEVELOPMENT USING WEB FRAMEWORK

L T P

2 - 4

RATIONALE

This course will cover the practical aspects of Web App development using various frameworks. The course equips students with resources for design, development and production of web applications. Students will be introduced to popular web application frameworks for building scalable web applications.

The following are the main objectives for this course:

- Motivate student's interest in learning Web-app development by giving them an insight into its possibilities through practical applications.
- Provide a sufficiently broad but practical introduction to Server-side web technologies..

LEARNING OUTCOMES

After undergoing the subject, the students will be able to:

- Identify basic aspects of web-frameworks.
- Apply the basic concepts, principles and practices of Web-site development using server-side technologies (PHP & MySQL)
- Install Word Press
- Create and manage Blogs, Websites using WordPress.
- Use PHP & MySQL with WordPress
- Install Moodle framework
- Create Web Application using Moodle
- Manage features of Moodle sites : Accounts, Enrollment, Roles & Permission
- Take backup of site content

DETAILED CONTENTS

1. **Practice on various Web Technologies** (8 Periods)
Practice on HTML, CSS, Java Script, Ajax. Practice on Server side scripting i.e. PHP & MySQL, What is WordPress
2. **Introduction to Web frameworks** (2 Periods)
Introduction to WordPress, How WordPress Works, Introduction to Moodle, Pedagogy, Moodle site - basic structure

3. **WordPress** (10 Periods)

Installation of WordPress, Introduction to Blogging, Creating Blogs, Using Images, Wrapping Text Around Images, Comments, Post Formats, Linking to Posts, Pages, and Categories, Using Smilies, Links Manager, WordPress Feeds, Customizing Feeds, Use Gravatars in WordPress, Writing Code in Your Posts, Using Password Protection, Developing a Colour Scheme, Designing Headers, CSS Horizontal Menus, Dynamic Menu Highlighting, Navigation Links, Next and Previous Links, Styling for Print, Designing Your Post Meta Data Section, Separating Categories in your Post Meta Data Section, Customizing the Read More, Formatting Date and Time, Finding CSS Styles, Creating Individual Pages, Uploading Files, Using WordPress Themes, Templates, Template Tags, Template Hierarchy, Validating a Website, Know Your Sources, WordPress Site Maintenance

4. **Moodle** (12 Periods)

Installing Moodle, Installing plugins, Version Upgrading, Verify Database Schema, Managing a Moodle site, Managing authentication, Manual accounts, No login, Email-based self-registration, IMAP authentication, Browse list of users, Bulk user actions, Add a new user, Upload users, User profile, Managing enrolment plugins, Manual enrolment, Guest access, Category enrolments, External database enrolment, Managing Roles and permissions, Assign roles, Site administrator, Manager role, Course creator role, Teacher role, Non-editing teacher role, Student role, Guest role, Roles settings, Permissions, Password salting.

Site appearance, Front page, Front page settings, My Moodle, User profiles, Navigation, Course list, Themes, Theme settings, Standard themes, Installing a new theme, Header and footer, Language settings, Language customization, Server settings, System paths, Session handling, HTTP, Maintenance mode, Cleanup, Environment, Site registration, Config changes report, Using web services, Publishing a course, Blogs, Blog settings, Using Blogs, Comments, Tags, RSS feeds settings, Using RSS feeds, Using Calendar, Site backup, Course backup, Course restore, Automated course backup.

INSTRUCTIONAL STRATEGY

Since the entire course content is web based, students can practice it online. The teachers should have practice on this framework. Entire course is hands-on based so practicals should be conducted in the laboratory.

LIST OF PRACTICALS

1. Practice on HTML, CSS, Java Script, Ajax.PHP & MySql
2. Install WordPress & Create Blogs
3. Manage blogs features e.g. Images, Text Around Images, Comments, Post Formats, Linking, Pages, Categories, Smilies, Feeds, Gravatars, Password Protection
4. Practice various designing features: Colour Scheme, Headers, CSS Horizontal Menus, Dynamic Menu, Highlighting, Navigation Links, Print

5. Read More, Formatting Date and Time, Finding CSS Styles, Creating Individual Pages, Uploading Files, Using WordPress Themes, Templates, Template Tags, Template Hierarchy, Validating a Website, Know Your Sources, WordPress Site Maintenance
6. Integrate PHP & MySql with WordPress
7. Install Moodle & various plugins,
8. Create a Moodle site and Database Schema
9. Design Site appearance, Front page, Front page settings, My Moodle, User profiles, Navigation, Course list, Themes, Theme settings, Header and footer, Language settings, Using web services, Publishing a course, Blogs, RSS feeds
10. Manage Moodle site, Managing authentication, Manual accounts, No login, Email-based self-registration, Account
11. Create Roles and permissions, Assign roles,
12. Implement Password salting.
13. Perform Site backup, Course backup, Course restore, Automated course backup

RECOMMENDED RESOURCES

1. <http://www.wpbeginner.com/beginners-guide/how-to-learn-wordpress-for-free-in-a-week-or-less/>
2. https://docs.moodle.org/22/en/Table_of_Contents

SUGGESTED DISTRIBUTION OF MARKS

| Topic No. | Time Allotted (Hrs) | Marks Allotted (%) |
|--------------|---------------------|--------------------|
| 1. | 8 | 18 |
| 2. | 2 | 10 |
| 3. | 10 | 16 |
| 4. | 12 | 28 |
| Total | 32 | 100 |

6.2 ARTIFICIAL INTELLIGENCE

L T P

4 - 3

RATIONALE

To be familiar with the applicability, strengths, and weaknesses of the basic knowledge representation, problem solving, machine learning, knowledge acquisition and learning methods in solving particular engineering problems

LEARNING OUTCOMES

On completion of this course, the students will be able to

1. Learn the basics and applications of artificial intelligence and categorize various problem domains, basic knowledge representation and reasoning methods.
2. Analyze basic and advanced search techniques including game playing, evolutionary search algorithms, constraint satisfaction.
3. Learn and design intelligent agents for concrete computational problems.
4. Design of programs in AI language(s).
5. Acquire knowledge about the architecture of an expert system and design new expert systems for real life applications.

DETAILED CONTENTS

1: Introduction- What is intelligence? Foundations of artificial intelligence (AI). History of AI; Problem Solving- Formulating problems, problem types, states and operators, state space, search strategies; Goals Of AI , Stages of AI ,Types of Artificial Intelligence;

(10 hrs)

2:- Informed Search Strategies- Best first search, A* algorithm, heuristic functions, Iterative deepening A*(IDA), small memory A*(SMA); Game playing - Perfect decision game, imperfect decision game, evaluation function, alpha-beta pruning.

(8 hrs)

3: Reasoning-Representation, Inference, Propositional Logic, predicate logic (first order logic), logical reasoning, forward chaining, backward chaining; AI languages and tools - Lisp, Prolog, CLIPS

(10 hrs)

4: Planning- Basic representation of plans, partial order planning, planning in the blocks world, hierarchical planning, conditional planning, representation of resource constraints, measures, temporal constraints

(8 hrs)

5: Uncertainty - Basic probability, Bayes rule, Belief networks, Default reasoning, Fuzzy sets and fuzzy logic; Decision making- Utility theory, utility functions, Decision theoretic expert systems. (12 hrs)

6: Inductive learning - Overview of Neural Network, Machine Learning and Deep Learning. Difference Between NLP, AI, ML, DL & NN ,Machine Learning Algorithms: decision trees, rule based learning, current-best-hypothesis search, least- commitment search; Reinforcement learning, Genetic algorithms; Deep learning Algorithms: Convolutional Neural Network, Recurrent Neural Network. (16 hrs)

LIST OF PRACTICALS:

1. Study of PROLOG: Write the following programs using PROLOG
2. Write a program to solve 8 queens problem
3. Solve any problem using depth first search.
4. Solve any problem using breadth first search.
5. Solve any problem using best first search.
6. Solve 8-puzzle problem using best first search
7. Solve traveling salesman problem.
8. Building Neural Network Model using Tensorflow.
9. Building Neural Network Model using Pytorch.
10. Building Basic Deep Learning Model with Keras
11. Building a Cat/Dog Classifier using a Convolutional Neural Network.

LIST OF RECOMMENDED BOOKS

1. Stuart Russell and Peter Norvig. Artificial Intelligence – A Modern Approach, Pearson Education Press, 2001.
2. Kevin Knight, Elaine Rich, B. Nair, Artificial Intelligence, McGraw Hill, 2008.
3. George F. Luger, Artificial Intelligence, Pearson Education, 2001.
4. D.W.Patterson, Artificial Intelligence And Expert Systems, Prentice Hall
5. Nils J. Nilsson, Artificial Intelligence: A New Synthesis, Morgan Kauffman, 2002.

SUGGESTED DISTRIBUTION OF MARKS

| Topic No. | Time Allotted (Hrs) | Marks Allotted (%) |
|------------------|----------------------------|---------------------------|
| 1. | 10 | 20 |
| 2. | 8 | 10 |
| 3. | 10 | 15 |
| 4. | 8 | 10 |
| 5. | 12 | 25 |
| 6. | 16 | 20 |
| Total | 64 | 100 |

6.3 CLOUD COMPUTING

L T P
2 - 4

RATIONALE

This course offers a good understanding of cloud computing concepts and challenges faced in implementation of cloud computing.

LEARNING OUTCOMES

After undergoing the subject, a student would be able to:

- Explain core concepts of cloud computing paradigm.
- Explain various Service Models
- Explain various Deployment Models.
- Describe SLA management in Cloud Computing
- Explain and apply the concept of virtualization.
- Describe the scheduling of tasks in cloud.
- Illustrate the fundamental concepts of cloud storage.
- Describe various security issues in the cloud.
- Use cloud computing

DETAILED CONTENTS

1. Introduction (6 hrs)

Evolution of Cloud Computing, Cloud Computing Overview, Characteristics, Applications, Benefits, Challenges.

2. Service and Deployment Models (6 hrs)

- Cloud Computing Service Models: Infrastructure as a Service, Platform as a Service, Software as a Service;
- Cloud Computing Deployment Models: Private Cloud; Public Cloud, Community Cloud, Hybrid Cloud, Major Cloud Service providers.

3. Service Level Agreement (SLA) Management (4 hrs)

Overview of SLA, Types of SLA, SLA Life Cycle, SLA Management Process.

4. Virtualization Concepts (6 hrs)
Overview of Virtualization, Types of Virtualization, Benefits of Virtualization, Hypervisors.
5. Cloud Security (4 hrs)
Infrastructure Security, Data Security & Privacy Issues, Legal Issues in Cloud Computing.
6. Cloud Storage (4 hrs)
Overview; Storage as a Service, Benefits and Challenges, Storage Area Networks(SANs).
7. Scheduling in Cloud (4 hrs)
Overview of Scheduling problem, Different types of scheduling, Scheduling for independent and dependent tasks, Static vs. Dynamic scheduling.

LIST OF PRACTICALS

1. Case study on amazon cloud service provider
2. Introduction and installation of CloudSim package
3. To study about configuration of Cloud Broker, Data center, VMs etc.
4. Installing VMWARE workstation and creating a virtual machine
5. Implementing any Scheduling algorithm

RECOMMENDED BOOKS

1. Rajkumar Buyya, James Broberg, Andrzej Goscinski (Editors): Cloud Computing: Principles and Paradigms, Wiley, 2011
2. Kumar Saurabh, Cloud Computing, Wiley, 2012.
3. Barrie Sosinsky: Cloud Computing Bible, Wiley, 2011.
4. Judith Hurwitz, Robin Bloor, Marcia Kaufman, Fern Halper: Cloud Computing for Dummies, Wiley, 2010.

SUGGESTED DISTRIBUTION OF MARKS

| Topic No. | Time Allotted (Hrs) | Marks Allotted (%) |
|------------------|--------------------------------|-------------------------------|
| 7. | 06 | 20 |
| 8. | 05 | 15 |
| 9. | 04 | 10 |
| 10. | 06 | 20 |
| 11. | 04 | 10 |
| 12. | 04 | 10 |
| 13. | 03 | 15 |
| Total | 32 | 100 |

ELECTIVE -II**ELECTIVE****6.4 (a) OPEN SOURCE TECHNOLOGIES**

| | | |
|----------|----------|----------|
| L | T | P |
| 2 | - | 3 |

RATIONALE

To provide exposure in FOSS (Free & Open Source Software), Manage open source projects & create websites.

LEARNING OUTCOME

After undergoing the subject, students will be able to:

- Install and use open source software
- Develop web sites using content management systems
- Use version control systems
- Install and configure LAMP/WAMP Stack
- Write shell scripts

DETAILED CONTENTS:

- Introduction
Need of Open Sources – Advantages of Open Sources, OpenSource Applications, FOSS usage - Free Software Movement, Commercial aspects of Open Source movement, Open Source Software Development Model, Comparision with close source / Proprietary software, Widely used open source software licenses, Apache License, BSD license, GNU General Public License, GNU Lesser General Public License, MIT License, Eclipse Public License and Mozilla Public License.
(7 hrs)
- Git and version control, Getting oriented with Git, Making and using a Git repository, Using Git with a GUI, Tracking and updating files in Git, Committing parts of changes, Cloning, Collaborating with remotes, Pushing your changes, Working with GitHub.
(10 hrs)
- Introduction to Content Management System. Various CMS Available, Installing Wordpress, Using Wordpress, Creating Web Site using Wordpress. Installing Plugins & Themes.
(7 hrs)
- Introduction to LAMP/WAMP Stack. Installing LAMP/WAMP Stack. Introduction of Linux, Apache, Mysql & PHP
(04 hrs)
- Basic Shell Scripting , Executing Script, Working with Variables and Input, Using Control Structure
(04 hrs)

LIST OF PRACTICALS

1. Installation of WAMP/LAMP/XAMP server
2. Configure Git
3. Setting up account on github
4. Creating and cloning repository
5. Installing Wordpress
6. Creating menus in Wordpress
7. Write a simple shell script that takes any number of arguments on the command line, and prints the arguments with “Hello” in front.
8. Write a simple shell script that takes two numbers as parameters and uses a while loop to print all the numbers from the first to the second inclusive, each number separated only by a space from the previous number.

RECOMMENDED BOOKS

1. 1. Joseph Feller, Brian Fitzgerald, Scott A. Hissam and Karim R. Lakhani Perspectives On Free And Open Source Software ISBN: 9780262562270 MIT Press
2. Pro Git Paperback by [Scott Chacon](#) (Author), [Ben Straub](#) (Author)
3. Git Pocket Guide Paperback – by [Richard E. Silverman](#) (Author)
4. Wordpress Websites Step-by-Step: The Complete Beginner's Guide to Creating a Website or Blog With Wordpress Paperback – Import, 4 Feb 2014 by [Caimin Jones](#)
5. J. Feller and B. Fitzgerald, Understanding Open Source Software Development, Addison Wesley, 2002.
6. Karl Fogel and Moshe Bar, Open Source Development with CVS, Third Edition, URL: <http://cvsbook.red-bean.com>
7. [www. sourceforge.net](http://www.sourceforge.net)
8. freeopensourcesoftware.org/
9. Philosophy of GNU URL: <http://www.gnu.org/philosophy/>
10. Version control system , URL: <http://git-scm.com/>
11. Linux Shell scripting Cookbook: Sarath Lakshman

SUGGESTED DISTRIBUTION OF MARKS

| Topic | Time Allotted Hrs) | Marks Allotted (%) |
|--------------|--------------------|--------------------|
| 1 | 7 | 25 |
| 2 | 10 | 30 |
| 3 | 7 | 25 |
| 4 | 04 | 10 |
| 5 | 04 | 10 |
| Total | 32 | 100 |

6.4 (b) COMPUTER PROGRAMMING USING PYTHON

L T P
2 - 3

RATIONALE

This course introduces the student to the Python language. Upon completion of this class, the student will be able to write non trivial Python programs dealing with a wide variety of subject matter domains.

Topics include language components, the IDLE environment, control flow constructs, strings, I/O, collections, classes, modules, and regular expressions

Course Outcomes

After undergoing the subject, the students will be able to:

- Execute Python code in a variety of environments
- Use correct Python syntax in Python programs
- Use the correct Python control flow construct
- Write Python programs using various collection data types
- Write home grown Python functions
- Use many of the standard Python modules such as os, sys, math, and time
- Trap various errors via the Python Exception Handling model
- Use the IO model in Python to read and write disk files
- Create their own classes and use existing Python class. Understand and use the Object Oriented paradigm in Python programs
- Use the Python Regular Expression capabilities for data verification

DETAILED CONTENTS

1. Introduction (03 hrs)

- A Brief History of Python
- Python Versions
- Installing Python
- Environment Variables
- Executing Python from the Command Line
- IDLE
- Editing Python Files
- Python Documentation
- Getting Help
- Dynamic Types
- Python Reserved Words
- Naming Conventions

2. Basic Python Syntax (02 hrs)
 - Basic Syntax
 - Comments
 - String Values
 - String Methods
 - The format Method
 - String Operators
 - Numeric Data Types
 - Conversion Functions
 - Simple Output
 - Simple Input
 - The % Method
 - The print Function
3. Language Components (06 hrs)
 - Indenting Requirements
 - The if Statement
 - Relational and Logical Operators
 - Bit Wise Operators
 - The while Loop
 - break and continue
 - The for Loop
4. Collections (04 hrs)
 - Introduction
 - Lists
 - Tuples
 - Sets
 - Dictionaries
 - Sorting Dictionaries
 - Copying Collections
 - Summary
5. Functions (03 hrs)
 - Introduction
 - Defining Your Own Functions
 - Parameters
 - Function Documentation
 - Keyword and Optional Parameters
 - Passing Collections to a Function
 - Variable Number of Arguments

- Scope
 - Functions - "First Class Citizens"
 - Passing Functions to a Function
 - map
 - filter
 - Mapping Functions in a Dictionary
 - Lambda
 - Inner Functions
 - Closures
6. Modules (02 hrs)
- Modules
 - Standard Modules - sys
 - Standard Modules - math
 - Standard Modules - time
 - The dir Function
7. Exceptions (02 hrs)
- Errors
 - Runtime Errors
 - The Exception Model
 - Exception Hierarchy
 - Handling Multiple Exceptions
 - Raise
 - assert
8. Input and Output (02 hrs)
- Introduction
 - Data Streams
 - Creating Your Own Data Streams
 - Access Modes
 - Writing Data to a File
 - Reading Data From a File
 - Additional File Methods
 - Using Pipes as Data Streams
 - Handling IO Exceptions
9. Classes in Python (05 hrs)
- Classes in Python
 - Principles of Object Orientation

- Creating Classes
 - Instance Methods
 - File Organization
 - Special Methods
 - Class Variables
 - Inheritance
 - Polymorphism
10. Regular Expressions (03 hrs)
- Introduction
 - Simple Character Matches
 - Special Characters
 - Character Classes
 - Quantifiers
 - The Dot Character
 - Greedy Matches
 - Grouping
 - Matching at Beginning or End
 - Match Objects
 - Substituting
 - Splitting a String
 - Compiling Regular Expressions
 - Flags

LIST OF PRACTICALS

1. Getting started with Python and IDLE in interactive and batch modes
2. What do the following string methods do?
 - lower
 - count
 - replace
3. Write instructions to perform each of the steps below
 - (a) Create a string containing at least five words and store it in a variable.
 - (b) Print out the string.
 - (c) Convert the string to a list of words using the string split method.
 - (d) Sort the list into reverse alphabetical order using some of the list methods (you might need to use dir(list) or help(list) to find appropriate methods).
 - (e) Print out the sorted, reversed list of words.
4. Write a program that determines whether the number is prime.
 What is your favorite number? 24
 24 is not prime

What is your favorite number? 31

31 is prime

5. Find all numbers which are multiple of 17, but not the multiple of 5, between 2000 and 2500?
6. Swap two integer numbers using a temporary variable. Repeat the exercise using the code format: a, b = b, a. Verify your results in both the cases.
7. Find the largest of n numbers, using a user defined function largest().
8. Write a function myReverse() which receives a string as an input and returns the reverse of the string.
9. Check if a given string is palindrome or not.

INSTRUCTIONAL STRATEGY

Teachers should lay emphasis on practicals and experts from industries may be invited to deliver lectures and share experiences with the students.

RECOMMENDED BOOKS

1. Learning Python by Mark Lutz; Pratham Books, Bangalore
2. Foundations of Python Network Programming by John Goerzen and Brandeu Rhodes; Apress-eBook distributed by Springer Science and Business Media, New York
3. Dive Into Python by Mark Pilgrim; Pratham Books, Bangalore
4. Think Python by Allen B. Downey; O'Reilly Media
5. Python Programming For Beginners: A Must Read Introduction to Python Programming by Robert Richards; Pratham Books, Bangalore

SUGGESTED DISTRIBUTION OF MARKS

| Topic No. | Time Allotted (Hrs) | Marks Allotted (%) |
|--------------|---------------------|--------------------|
| 1. | 03 | 10 |
| 2. | 02 | 07 |
| 3. | 06 | 18 |
| 4. | 04 | 14 |
| 5. | 03 | 09 |
| 6. | 02 | 06 |
| 7. | 02 | 06 |
| 8. | 02 | 06 |
| 9. | 05 | 15 |
| 10. | 03 | 09 |
| Total | 32 | 100 |

6.4 (c) DEVELOPMENT OF ANDROID APPLICATIONS

L T P

2 - 3

RATIONALE

This course provides a basic understanding of Android development, including the use of content providers, creating audio and video services. This course focuses on helping people become an Android application developer and releasing high-quality apps to the marketplace. Learn about the various stages of development on the Android platform and study topics related to UI, application services, permissions and security, graphics and video resources, data persistence, monitoring tools, mobile app marketing, application hosting and more.

LEARNING OUTCOMES

On completion of this course, the students will be able to

- gain knowledge of set up and installation of Android
- gain App development knowledge
- gain knowledge of user interfaces on Mobile Apps.

DETAILED CONTENTS

1. Introduction to Android

(4 hrs)

What is Android? Setting up development environment, Dalvik Virtual Machine & .apk file extension , Fundamentals: Basic Building blocks - Activities, Services, Broadcast Receivers & Content providers , UI Components - Views & notifications , Components for communication -Intents & Intent Filters, Android API levels (versions & version names)

2. Installation and Setup on Android

(3 hrs)

Environment Setup – Installation & Setup of SDK tools on Windows; Installing platforms and samples; Creating an Android Virtual Device (emulator); Installing Eclipse on a Windows machine; Installing the Android Development Tools; Preparing an Android device for development.

3. Application Structure

(6 hrs)

AndroidManifest.xml, Uses-permission & uses-sdk, Resources & R.java, Assets, Layouts & Drawable Resources, Activities and Activity lifecycle, First sample Application

4. User interface and Controls (7 hrs)

Understanding units and layout; Using layout managers; Working with text controls; Building button controls; Building list controls; Building custom list layouts; Other interesting controls.

5. Emulator-Android Virtual Device (6 hrs)

Launching emulator, Editing emulator settings, Emulator shortcuts, Logcat usage, Introduction to DDMS, Hello World App, Creating your first project The manifest file Layout resource Running your app on Emulator.

6. Graphics and Animation (6 hrs)

Creating and using styles; Creating and using themes ; Creating icons; Creating NinePatch drawables, Setting up frame-by-frame animation; Showing tween animation; Working in 2D graphics.

LIST OF PRACTICALS

1. Installation of android studio
2. Create “Hello world “ program & run it on emulator
3. To work with Text view elements
4. To implement Calculator in android app
5. Program to create icons
6. To develop interactive UI
7. To work with Photo Gallery in android
8. To develop an web app

RECOMMENDED BOOKS

1. Mobile Apps for Android (IBM ICE).
2. Pro Android 2 -Sayed Hashimi | Satya Komatineni | Dave MacLean, Apress publication
3. The Busy Coder’s guide to Android Development By Mark L. Murphy
4. David Tainar – Mobile Computing: Concepts Methodologies, Tools & Applications.
5. Barbara L Ciarantaro – Mobile technology consumption.

SUGGESTED DISTRIBUTION OF MARKS

| Topic | Time Allotted Hrs) | Marks Allotted (%) |
|--------------|---------------------------|---------------------------|
| 1 | 4 | 15 |
| 2 | 3 | 10 |
| 3 | 6 | 20 |
| 4 | 7 | 25 |
| 5 | 6 | 15 |
| 6 | 6 | 15 |
| Total | 32 | 100 |

6.5 MAJOR PROJECT WORK

L T P
- - 12

RATIONALE

Major Project Work aims at developing innovative skills in the students whereby they apply in totality the knowledge and skills gained through the course work in the solution of particular problem or by undertaking a project. The individual students have different aptitudes and strengths. Project work, therefore, should match the strengths of students. For this purpose, students should be asked to identify the type of project work, they would like to execute. It is also essential that the faculty of the respective department may have a brainstorming to identify suitable project assignments for their students. The project assignment can be individual assignment or a group assignment. There should not be more than 3 students if the project work is given to a group. The students should identify themselves or accept the given project assignment at least two to three months in advance. The project work identified in collaboration with industry should be preferred. Each teacher is expected to guide the project work of 5–6 students.

The project assignments may consist of:

- Web page designing including database connectivity
- mobile applications designed using android
- Programming customer based applications
- Database applications
- Software Development
- algorithm designing to improve complexity
- projects related to clouds
- Bringing improvements in the existing systems/equipment
- Projects related to Multimedia
- Projects related Application development using web framework and android
- Web Hosting
- Software development using Python

- Configuration of Network Operating System(Windows, Linux)

A suggestive criterion for assessing student performance by the external (personnel from industry) and internal (teacher) examiner is given in table below:

| Sr. No. | Performance criteria | Max.* marks | Rating Scale | | | | |
|--------------------|---|-------------|--------------|-----------|-----------|-----------|-----------|
| | | | Excellent | Very Good | Good | Fair | Poor |
| 1. | Selection of project assignment | 10 | 10 | 8 | 6 | 4 | 2 |
| 2. | Planning and execution of considerations | 10 | 10 | 8 | 6 | 4 | 2 |
| 3. | Quality of performance | 20 | 20 | 16 | 12 | 8 | 4 |
| 4. | Providing solution of the problems or production of final product | 20 | 20 | 16 | 12 | 8 | 4 |
| 5. | Sense of responsibility | 10 | 10 | 8 | 6 | 4 | 2 |
| 6. | Self expression/ communication skills | 5 | 5 | 4 | 3 | 2 | 1 |
| 7. | Interpersonal skills/human relations | 5 | 5 | 4 | 3 | 2 | 1 |
| 8. | Report writing skills | 10 | 10 | 8 | 6 | 4 | 2 |
| 9. | Viva voce | 10 | 10 | 8 | 6 | 4 | 2 |
| Total marks | | 100 | 100 | 80 | 60 | 40 | 20 |

The overall grading of the practical training shall be made as per following table

| | Range of maximum marks | Overall grade |
|------|------------------------|---------------|
| i) | More than 80 | Excellent |
| ii) | 79 > 65 | Very good |
| iii) | 64 > 50 | Good |
| iv) | 49 > 40 | Fair |
| v) | Less than 40 | Poor |

In order to qualify for the diploma, students must get “Overall Good grade” failing which the students may be given one more chance of undergoing 8 -10 weeks of project oriented professional training in the same industry and re-evaluated before being disqualified and declared “not eligible to receive diploma”. It is also important to note that the students must get more than six “goods” or above “good” grade in different performance criteria items in order to get “Overall Good” grade.

Important Notes

- 1. This criteria must be followed by the internal and external examiner and they should see the daily, weekly and monthly reports while awarding marks as per the above criteria.**
- 2. The criteria for evaluation of the students have been worked out for 100 maximum marks. The internal and external examiners will evaluate students separately and give marks as per the study and evaluation scheme of examination.**
- 3. The external examiner, preferably, a person from industry/organization, who has been associated with the project-oriented professional training of the students, should evaluate the students performance as per the above criteria.**
4. It is also proposed that two students or two projects which are rated best be given merit certificate at the time of annual day of the institute. It would be better if specific nearby industries are approached for instituting such awards.

The teachers are free to evolve another criteria of assessment, depending upon the type of project work.

The students must submit a project report of not less than 50 pages (excluding coding). The report must follow the steps of Software Engineering Concepts

It is proposed that the institute may organize an annual exhibition of the project work done by the students and invite leading Industrial organizations in such an exhibition. It is also proposed that two students or two projects which are rated best be given merit certificate at the time of annual day of the institute. It would be better if specific industries are approached for instituting such awards.