CODE	COURSE NAME	CATEGORY	L	T	P	CREDIT
20MCA245	MINI PROJECT	PROJECT	-	-	4	2

Preamble: This project work aims to enable the students to apply the software engineering principles on a real software project, to make the students familiar with the stages of a deployment pipeline and to develop a software product using the latest software development methodology.

Prerequisite: Knowledge in software engineering principles and programming skills.

Course Outcomes: After the completion of the course the student will be able to

CO No.	Course Outcome (CO)	Bloom's Category Level
CO 1	Identify a real-life project which is useful to society / industry	Level 2: Understand
CO 2	Interact with people to identify the project requirements	Level 3: Apply
CO 3	Apply suitable development methodology for the development of the product / project	Level 3: Apply
CO 4	Analyse and design a software product / project	Level 4: Analyse
CO 5	Test the modules at various stages of project development	Level 5: Evaluate
CO 6	Build and integrate different software modules	Level 6: Create
CO 7	Document and deploy the product / project	Level 3: Apply

Mapping of course outcomes with program outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	2	3	3	3	1	2	3	3	3	3	3	3
CO 2	2	3	2	3	2	3	2	1	3	2	3	
CO 3	3	3	3	3	3	1	3	3	1		2	
CO 4	3	3	3	3	3		3	3	1	1	2	
CO 5	3	3	3	3	3		2	3			1	
CO 6	3	3	3	3	3	2	3	3		2	3	3
CO 7	1	1	3	3	3	2	3	3	2	1	2	

3/2/1: High/Medium/Low

Mark distribution

Total Marks	CIE	ESE
100	100	-

Assessment Criteria

Class participation and attendance	10%
Evaluation	50%
Class work	40%

Marks Division

Continuous evaluation by Supervisor, Scrum Master and Project Guide	50 Marks
Interim evaluation by the Project Assessment Board	25 Marks
Final evaluation by the Project Assessment Board	25 Marks
Total	**100 Marks**

Guidelines:

● Students shall identify Real-Life Projects which are relevant and useful to the society or industry.

● The project shall be an individual project and must be done in-house. The student has to spend time in the lab for the project work.

● Attendance as per MCA regulations is applicable for submitting the project for final evaluation.

● Students shall submit project synopsis and get prior approval from the Project (Faculty) Supervisor before the project work begins.

● If there is a customer for the project then he/she will be the Product Owner (External Guide) and a faculty from the department will be the Internal Guide. If there is no such customer then the Internal Guide himself/herself shall act as the Product Owner.

● A faculty / technical staff shall act as the Scrum Master to continuously monitor the project development. Periodic meetings, of less than 15 minutes, at the convenience of

the Scrum Master are to be highly encouraged. Ensure such meetings occur once in three days.

● Set a sprint as two weeks, ensure biweekly reviews. A review shall not exceed 30 minutes. A demo to the Product Owner (Project Guide) is mandatory in every review.

● The student shall maintain a Scrum Book (Rough Record) which has to be divided into 4 parts – (i) Product Backlog (ii) Database & UI Design (iii) Testing & Validation and (iv) Details of Versions. Make dated entries in the corresponding part at regular intervals. The corrections and comments from Product Owner and Scrum Master should be clearly indicated with the Date.

● Test Driven Development methodology may be practiced for the project development. BugZilla, BackLog or any such tool may be used for Bug Tracking.

● Git shall be used for Version Control and Git commit history may be verified as part of project evaluation .

● LaTeX or an equivalent tool shall be used for preparing Presentations and Project Report.

● Interim evaluations of project's progress shall be conducted as part of Internal Assessment. Project Evaluation Board may consist of Project Supervisor, Product Owner, Scrum Master and one other Faculty Member from the department. Scrum reviews shall not be sacrificed for such presentations.

● At the end of the semester entire project development activities shall be evaluated internally by the Project Evaluation Board.

Week	Schedule
1	Familiarisation with build tools (editor/IDE, compiler such as gcc with commonly used options/switches, debugger like dgb). Familiarisation with an IDE (Eclipse, NetBeans…), that supports build tools and common version control operations using Git . Familiarisation with Docker Selection of Topic, Formation of Development Team, Feasibility analysis.
2	Topic Approval, Meeting of Development Team including Scrum Master with Product Owner. Informal, preliminary discussions of requirements. Creating user stories in the rough record.

3	Commencement of the Project.
	Identifying modules, Initial Design of Database & UI.
	Creating a Docker container for the environment
	Creating an empty git repository by Scrum Master / one member of the Development team and setting permission to other members.
	Pushing the first version of the Project along with a Readme file containing contact details of team members.
	Creating pull requests for sample update of Readme by each member and merging the pull requests of one by another.
4-5	Setting up systems for development, testing and production.
	Design of the basic model of a simple deployment pipeline
	Creating a suitable folder structure (Maven's folder structure is desirable). Creating Unit tests using an XUnit framework, Writing the build and code analysis script, Writing acceptance test scripts and test cases, Setting up a Continuous Integration System like Jenkins. Automating acceptance tests with Selenium, Karate or an equivalent tool, writing a simple deployment script that uses scp/rsync or Ansible for copying the Dockerfile and running Docker with ssh.
	First Scrum Review.
	(Here onwards, the Scrum reviews are conducted on every other week)
7	Project Presentation - Interim
	Evaluation to be based on Git History
14	Submission of Project Report, with Scrum Book
	Project Presentation – Final
	Evaluation to be based on Git History, Scrum Book, Project Report and Presentation

References

1. Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation (Addison-Wesley Signature Series (Fowler)) 1st Edition

2. Alistair Cockburn, Agile Software Development: The Cooperative Game, Addison Wesley, 2nd Edition (2006).

3. Andrew Hunt, David Thomas, The Pragmatic Programmer: From Journeyman to Master, Pearson India, 1st Edition (2008).

4. Ken Schwaber, Mike Beedle, Agile Software Development with Scrum, Pearson (2008).

5. Lisa Crispin, Janet Gregory, Agile Testing: A Practical Guide for Testers and Agile Teams, Addison Wesley Professional, 1st Edition (2008).

6. Mike Cohn, User Stories Applied: For Agile Software Development, Addison Wesley, 1st Edition, (2004).

7. Pressman, R.S., Software Engineering: A Practitioner's Approach, McGraw Hill SE, 7th Edition, (2010).

8. Robert C. Martin, Agile Software Development, Principles, Patterns and Practices, Prentice Hall Imprint, Pearson Education, 2nd Edition (2002).\

9. Rod Stephens, □Beginning Software Engineering, Wrox Series, Wiley India Pvt Ltd (2015).

10. RyPress Ry's Git Tutorial (Free e-book)

Web Reference

1. Introduction to DevOps (https://www.edx.org/course/introduction-devops-microsoft-dev212x)