MAHATMA GANDHI UNIVERSITY
Bachelor in Computer Applications (Honours)
SYLLABUS MGU-BCA (Honours)

Sixth Semester								
Course Code	Course Type	Course Title	Hours / week	*L	*T	* P	*O	Credit
MG6CCRBCA300	CC	Generative AI	5	3	0	2	0	4
MG6SECBCA300	SEC	Software Development Lab	8	0	0	8	0	4
		#Professional Elective – VI & VII						
MG6DSEBCA300	DSE	Big Data Analytics (*Data Science Specialization*)	6	4	0	2	0	5
MG6DSEBCA301		Exploratory Data Analysis (*Data Science Specialization*)	6	4	0	2	0	5
MG6DSEBCA302		Deep Learning for Computer Vision (*Artificial Intelligence & Machine Learning Specialization*)	6	4	0	2	0	5
MG6DSEBCA303		Predictive Analytics (*Artificial Intelligence & Machine Learning Specialization*)	6	4	0	2	0	5
MG6DSEBCA304		DevOps Fundamentals with Docker and Jenkins (*Full Stack Development Specialization*)	6	4	0	2	0	5
MG6DSEBCA305		Web Security and Penetration Testing with Kali (*Full Stack Development Specialization*)	6	4	0	2	0	5
		TOTAL	25	11	0	14	0	18

*L-Lecture; *T-Tutorial; *P-Practical/Practicum; *O- Others
The student can choose any two DSE courses from the available options in the sixth semester.

MGU BCA (HONOURS) Syllabus

	Mahatma Gandhi University **Kottayam**
Programme	**BCA (Honours)**
Course Name	**Generative AI**
Type of Course	**Core Course**
Course Code	**MG6CCRBCA300**
Course Level	**NA**
Course Summary	This course introduces the fundamental concepts, models and applications of Generative Artificial Intelligence, covering AI, Machine Learning, Deep Learning, GANs, VAEs, Diffusion Models, Transformers and Large Language Models. It explores prompt engineering, text and image generation, Retrieval-Augmented Generation (RAG), AI agents and practical applications across various domains. Students will gain hands-on experience using Python, Jupyter Notebook/Google Colab and Generative AI tools. The course also addresses ethical issues, limitations and responsible AI practices, equipping students with the knowledge and practical skills to use Generative AI effectively and responsibly.
Semester	6
Course Details	Learning Approach
Pre-requisites, if any	Basic knowledge of Python programming. Familiarity with machine learning concepts is desirable.

Credits	4			
Lecture	Tutorial	Practical	Others	Total Hours
3	0	1		75

MGU BCA (HONOURS) Syllabus

COURSE OUTCOMES (CO)

CO No.	Expected Course Outcome	Learning Domains *	PO No
1	Explain the fundamental concepts, architectures and models of Generative Artificial Intelligence.	U	1
2	Apply Generative AI tools and prompt engineering techniques to generate and process text, images and other digital content.	A	2
3	Analyse the applications, limitations, ethical issues and responsible-use practices associated with Generative AI.	An	2
4	Develop simple Generative AI applications using Python and appropriate AI tools.	A	2

Remember (K), Understand (U), Apply (A), Analyse (An), Evaluate (E), Create (C), Skill (S), Interest (I) and Appreciation (Ap)

COURSE CONTENT

Content for Classroom transaction (Units)

Module	Units	Course description	Hrs	CO No.
1		FUNDAMENTALS OF GENERATIVE AI		
	1.1	Introduction to Artificial Intelligence (AI), Machine Learning (ML), Deep Learning (DL) and Generative AI; evolution and development of Generative AI; differences between generative and discriminative AI; applications, advantages, limitations and challenges of Generative AI.	7	1
	1.2	Fundamentals of neural networks and deep learning; introduction to major generative architectures: Generative Adversarial Networks (GANs), Variational Autoencoders (VAEs) and Diffusion Models. Overview of Transformer architecture and attention mechanisms.	5	1
	1.3	Introduction to Large Language Models (LLMs), foundation models and popular Generative AI tools and platforms - ChatGPT, Google Gemini, Microsoft Copilot and Adobe Firefly.	3	1
2		LLM AND PROMPT ENGINEERING		
	2.1	Introduction to Natural Language Processing (NLP) and language modeling; tokens and tokenization; word and text embeddings; attention mechanisms and self-attention; Transformer architecture – encoder, decoder and encoder-decoder models.	5	2
	2.2	Fundamentals and working principles of Large Language Models; pre-training and fine-tuning; applications of LLMs. Prompt engineering – principles, techniques and best	5	2

MGU-BCA (HONOURS) Syllabus

		practices; zero-shot, one-shot and Chain of Thought (CoT) prompting, and few-shot prompting; prompt design, refinement and structured prompting.		
	2.3	Applications of LLMs: text generation, summarisation, translation, question answering and content creation. Structured JSON output;Introduction to Retrieval-Augmented Generation (RAG), embeddings and vector databases. Overview of text-to-image, text-to-audio and text-to-video generation.	5	2
3		APPLICATIONS, ETHICS AND RESPONSIBLE AI		
	3.1	Applications of Generative AI in education, healthcare, business, research and software development; AI-assisted coding, data analysis and document creation. Introduction to AI agents, tools and basic automation. Model customisation and fine-tuning – overview.	5	3
	3.2	Evaluation of Generative AI outputs: accuracy, relevance, reliability and consistency. Limitations of Generative AI; hallucinations and methods for verifying AI-generated information.	5	3
	3.3	Ethical and responsible use of Generative AI: bias, fairness, privacy, copyright and intellectual property rights. Deepfakes, misinformation, security risks and misuse of Generative AI. Responsible AI practices, transparency, human supervision and verification of AI-generated content.	5	3
		LAB ASSIGNMENTS		
		1. Inspecting tokens and token limits using online tokenizers; **2.** Experimenting with temperature and prompt settings in free browser playgrounds (such as Google AI Studio)		

MGU BCA (HONOURS) Syllabus

4		**3.** Practicing zero-shot, few-shot, and Chain-of-Thought prompts **4.** Exploring GAN based image generation and analysing generated outputs for fidelity, latent transitions and mode collapse. **5.** Extracting clean JSON data from unstructured text **6.** Setting up free API keys securely in Python **7.** Creating a simple command-line Q&A script **8.** Building a basic multi-turn chatbot that remembers conversation history **9.** Creating a text-file summarizer **10.** Testing and defending against prompt injection attacks **11.** Building an automated multiple-choice quiz generator.	30	4

Teaching and Learning Approach	**Classroom Procedure (Mode of transaction)** ● ICT Enabled lecture ● Interactive sessions ● Class discussions ● Lab exercise ● Hands-on demonstrations and project-based learning

MGU BCA (HONOURS)
Syllabus

Assessment Types	**MODE OF ASSESSMENT** **A. Continuous Comprehensive Assessment (CCA)** **CCA for Theory: 25 Marks** 1. Written test 2. Assignments 3. Quiz 4. Viva **CCA for Practical: 15 Marks** 1. Practical assignments 2. Lab Record 3. Observation of practical skills 4. Viva
	B. End Semester Evaluation **ESE for Theory: 50 Marks (1.5 Hrs)** Part A: Very Short Answer Questions (Answer all) - (10*2=20 Marks) Part B: Short Answer Questions (3 out of 5 Questions) - (3*5=15 Marks) Part C: Essay Questions (1 out of 2 Questions) - (1*15=15Marks) **ESE for Practical: 35 Marks (1.5 Hrs)** 1. Coding and Output - 20 Marks 2. Viva - 10 Marks 3. Record - 5 Marks

MGU-BCA (HONOURS)

REFERENCES

Syllabus

1. T V Geetha, Generative AI, Pearson Education, 2025

2. Numa Dhamani and Maggie Engler, Introduction to Generative AI, Manning Publications, 2024.

3. Jay Alammar and Maarten Grootendorst, Hands-On Large Language Models, O'Reilly Media, 2024.

4. Akshay Kulkarni, Adarsha Shivananda and Dilip Gudivada. Applied

5. Generative AI for Beginners, Apress, 2023.

SUGGESTED READING:

1. Sebastian Raschka, Build a Large Language Model (From Scratch), Manning Publications, 2024.
2. David Foster, Generative Deep Learning, 2nd Edition, O'Reilly Media, 2023.
3. Google AI for Developers — Gemini API and Google AI Studio documentation (ai.google.dev).
4. Hugging Face LLM Course (huggingface.co/learn).
5. Melanie Mitchell, Artificial Intelligence: A Guide for Thinking Humans, Farrar, Straus and Giroux, 2019.

MGU-BCA (HONOURS)
Syllabus

Mahatma Gandhi University
Kottayam

Programme	**BCA (Honours)**					
Course Name	**Software Development Lab**					
Type of Course	**SEC**					
Course Code	**MG6SECBCA300**					
Course Level	**NA**					
Course Summary	Each student shall undertake a project (individual project), following a systematic process that includes problem identification, analysis, design or methodology formulation, implementation or experimentation, testing or evaluation, documentation and final project report and presentation, as appropriate to the nature of the project.					
Semester	6	Credits			4	Total Hours
Course Details	Learning Approach	Lecture	Tutorial	Practical	Others	
		0	0	4	0	120
Pre-requisites, if any	Basic knowledge of programming concepts, literature review and problem-solving techniques. Understanding of data structures, database management systems, and software engineering fundamentals.					

MGU-BCA (HONOURS)

Syllabus

COURSE OUTCOMES (CO)

CO No.	Expected Course Outcome	Learning Domains *	PO No
1	Identify a real-world problem or research gap through domain study or literature review, and formulate a feasible project proposal with clearly defined objectives and scope.	An	1, 2, 4, 6, 9

2	Analyse project requirements and design appropriate solutions by developing system architecture, database schemas, user interfaces or research frameworks.	C	1, 2, 3
3	Evaluate the solutions using suitable tools, testing techniques and performance metrics.	E	3, 5, 10
4	Prepare project documentation and effectively present and demonstrate the project, adhering to professional and ethical practices.	S	4, 8, 10
Remember (K), Understand (U), Apply (A), Analyse (An), Evaluate (E), Create (C), Skill (S), Interest (I) and Appreciation (Ap)			

COURSE CONTENT

Content for Classroom transaction (Units)

Module	Units	Course description	Hrs	CO No.
1	**Phase 0**			
	1.1	**Domain Study and Problem Identification:** Introduction to project development and research-oriented projects, Domain study- study of existing systems or literature review (minimum of 10 papers); Identification of real-world problems and user requirements or finding the research gap; Feasibility study.	12	1
	1.2	**Project Proposal:** Clearly define the problem statement or research problem; Define project objectives and scope; Selection of development approaches or research methodologies; Identification of tools, technologies, programming languages and platforms; Specify expected outcomes and deliverable; Prepare a project work plan and timeline using a Gantt chart; Project proposal documentation and presentation. **Review 0:** Proposal presentation and approval.	12	1
2	**Phase 1**			

MGU BCA (HONOURS)

	2.1	**Requirement Engineering and Data Analysis:** Requirement gathering techniques – interviews, questionnaires, observation and document analysis; Identification and analysis of functional and non-functional requirements; Preparation of Software Requirement Specification (SRS); Preparation of use-case diagrams and DFDs. For research-oriented projects: dataset acquisition, data description, data preprocessing and exploratory data analysis; Identification of suitable baseline methods and performance parameters.	12	2
	2.2	**Design and Research Frame work:** System architecture design; database design – ER diagrams, relational schemas and normalisation; UML diagrams – class, sequence and activity diagrams; user interface and user experience design; API and module design. For research-oriented projects: formulation of research framework, selection of algorithms and techniques, experimental design, data analysis procedures and evaluation criteria; Preparation of system design documentation. **Review 1:** Presentation and evaluation of requirement analysis and system design.	12	2
3	**Phase 2.**			
	3.1	**Core Development and Implementation:** Set up the development environment and configuring required tools; selection of appropriate implementation strategies and follow appropriate coding standards, data structures and algorithms.; modular programming and	20	3

MGU-BCA (HONOURS) Syllabus

		reusable components; implementation of frontend and backend components; database connectivity; version control using Git/GitHub. For research-oriented projects: implementation of selected algorithms, models and techniques, data collection, preprocessing and preliminary experimental analysis.		
	3.2	**Integration and Iterative Development:** Integration of software modules and components; API development and integration with external services; database integration; implementation of authentication, authorisation and basic security measures; debugging, error handling and performance optimisation; iterative development and refinement based on feedback. For research-oriented projects: Refine the research methodology, experimental parameter tuning, conduct comparative analysis, validate findings and improve the proposed solution based on experimental observations. Project progress documentation and demonstration. **Review 2:** Project progress review – minimum 60% completion.	20	3
4	**Phase 3**			
	4.1	**Testing, Validation and Evaluation:** Test planning and test case preparation; functional and non-functional testing; unit testing, integration testing and system testing; software quality assessment based on functionality, reliability, usability and performance; For research-oriented projects: Evaluate the	24	3

MGU-BCA (HONOURS) Syllabus

		implemented methodologies using appropriate performance metrics, Analyse and interpret experimental results, comparative evaluation with existing methods or baseline approaches, identification of limitations and scope for improvement. **Review 3: Final internal Project demonstration.**		
5	**Final Project Report**			
	5.1	Preparation of project documentation including abstract, introduction, problem statement, objectives and scope, requirement analysis, system design, methodology, implementation, testing and evaluation, conclusion and future scope; For research-oriented projects: documentation of research methodology, datasets, experimental setup, algorithms, evaluation metrics, comparative analysis, research findings, limitations and reproducibility; **Final Submission:** Comprehensive project report, source code, datasets and supporting resources. Project presentation, demonstration and viva voce.	8	4

Teaching and Learning Approach	**Classroom Procedure /Mode of transaction** • Practical Sessions • Discussions • Presentations • Collaborative learning
Assessment Types	**MODE OF ASSESSMENT** **Continuous Comprehensive Assessment (CCA)** **CCA for Practical: 30 Marks**

Assessment Component	**Marks**

MGU-BCA (HONOURS) Syllabus

	Project Review 0 - Proposal presentation.	6
	Project Review 1 - Evaluation of requirement analysis and system design.	7
	Project Review 2- Progress review.	5
	Project Review 3- Final project demonstration.	5
	Documentation - Project Diary/Git Activity.	7
	End Semester Evaluation **ESE for Practical: (70 Marks)** Project Report - 20 Marks Presentation - 15 Marks Demonstration - 15 Marks Viva - 20 Marks	

MGU-BCA (HONOURS)

Syllabus

	Mahatma Gandhi University Kottayam					
Programme	**BCA (Honours)**					
Course Name	**Big Data Analytics**					
Type of Course	**DSE**					
Course Code	**MG6DSEBCA300**					
Course Level	**NA**					
Course Summary	This course introduces Big Data concepts and the Hadoop ecosystem, covering data classification, Hadoop features, HDFS, MapReduce and frameworks like Pig and Hive for Big Data applications. Students will gain practical experience with Hadoop tools and techniques for processing and analyzing large datasets.					
Semester	6	Credits			5	
Course Details	Learning Approach	Lecture	Tutorial	Practical	Others	Total Hours
		4	0	1	0	90
Pre-requisites, if any	Basic knowledge in Database Management Systems.					

COURSE OUTCOMES (CO)

CO No.	Expected Course Outcome	Learning Domains*	PO No
1	Illustrate the fundamental concepts, characteristics, challenges and applications of Big Data Analytics.	U	1,2
2	Understand the features, architecture and ecosystem of Hadoop, including HDFS and YARN.	U	1,2
3	Demonstrate the MapReduce programming model using Mapper, Reducer, Combiner and Partitioner for processing large datasets.	A	1,2,8
4	Compare Apache Pig and Hive based on their architecture, features, data types and applications.	An	1,2
5	Apply Pig Latin and HiveQL operations for processing and analysing large datasets.	A	1,2,8

Remember (K), Understand (U), Apply (A), Analyse (An), Evaluate (E), Create (C), Skill (S), Interest (I) and Appreciation (Ap)

MGU BCA (HONOURS) Syllabus

COURSE CONTENT

Content for classroom transaction (Units)

Module	Units	Course description	Hrs	CO No.
1		**Introduction to Big Data**		
	1.1	**Classification of Digital Data** Classification of Digital Data – Structured, Semi-structured and Unstructured Data, Characteristics of Data. **Definition and Evolution of Big Data** Characteristics of Big Data – Five Vs (Volume, Velocity, Variety, Veracity and Value), Why Big Data? Challenges with Big Data.	8	1
	1.2	**Big Data Processing and Applications** Applications of Big Data, Traditional Data Processing versus Big Data Processing, Traditional Business Intelligence (BI) versus Big Data, A Typical Data Warehouse Environment, Brief introduction to Hadoop Environment.	4	1
	1.3	**Introduction to Big Data Analytics** Definition and Importance of Big Data Analytics, Types, Classification of Analytics – Descriptive, Diagnostic, Predictive and Prescriptive Analytics, Challenges in Big Data Analytics, Data Science – Introduction, Role of a Data Scientist.	3	1
2		**Introduction to Hadoop**		
	2.1	**Features of Hadoop** Features, Key Advantages of Hadoop, Versions of Hadoop, Hadoop Use Cases **Hadoop Architecture and Ecosystem** Overview of Hadoop Ecosystem, Hadoop Architecture, Hadoop Distributions, Hadoop versus SQL, RDBMS versus Hadoop.	6	2
	2.2	**Hadoop Distributed File System (HDFS)** Introduction to HDFS, HDFS Architecture, HDFS Daemons, NameNode and DataNode, Secondary NameNode, Special	3	2

MGU-BCA (HONOURS)
Syllabus

		Features of HDFS, HDFS Blocks and Replication.		
	2.3	**Working with HDFS** Anatomy of File Read Operation, Anatomy of File Write Operation, Working with HDFS Commands, HDFS File System Operations, File Management using HDFS Commands.	3	2
	2.4	**Hadoop Resource Management – YARN** Introduction to YARN, Architecture of YARN, Components of YARN, ResourceManager, NodeManager, ApplicationMaster, Managing Resources with YARN.	3	2
3		**MapReduce Programming**		
	3.1	**Introduction to MapReduce** Introduction to MapReduce, Features of MapReduce, MapReduce Architecture, MapReduce Daemons, Working of MapReduce, MapReduce Working Example.	4	3
	3.2	**MapReduce Programming Model** MapReduce Programming, Mapper, Reducer, Combiner, Partitioner, Role of Mapper and Reducer in MapReduce.	3	3
	3.3	**MapReduce Job Execution** Anatomy of a MapReduce Job Run, Job Scheduling, Task Execution, Shuffle and Sort, MapReduce Job Execution Flow, Failures in MapReduce.	4	3
	3.4	**MapReduce Input and Output** MapReduce Input Types and Formats, MapReduce Output Types and Formats, Input Format and Output Format, MapReduce Applications.	4	3
4		**Big Data Processing Using Pig and Hive**		
	4.1	**Introduction to Apache Pig** Introduction to Pig, Applications of Big Data Using Pig, Pig Latin Overview, Pig Latin Data Types, Complex Data Types, Pig Latin Running Modes.	5	4

MGU-BCA (HONOURS) Syllabus

	4.2	**Pig Latin Operators** Pig Latin Operators, Relational Operators, AVG, MAX and COUNT Functions, Word Count Example Using Pig.	3	4
	4.3	**Introduction to Apache Hive** Introduction to Hive, Hive Architecture, Hive Data Types, Hive File Formats, HiveQL, Hive Applications **Comparative Study of Big Data Technologies** MapReduce versus Pig, Pig versus Hive	7	4
5	**LAB Questions**			
	5.1	1.Create a directory in HDFS, upload a text file into it, display the contents of the file, and list the files in the directory using appropriate HDFS commands.	30	5
		2.Write and execute HDFS commands to create, copy, move, rename, and delete files and directories.		
		3.Upload a sample text file from the local system to HDFS and then download the file from HDFS back to the local system.		
		4.Implement a simple MapReduce program to count the frequency of each word in a given text file.		
		5.Implement a MapReduce program to find the maximum number from a given set of numbers stored in a text file.		
		6.Write a MapReduce program to count the number of occurrences of each character in a given text file		
		7.Using Apache Pig, load a text file, split the contents into words, group the words, and find the frequency of each word using Pig Latin.		
		8..Create a sample student dataset in Pig and perform FILTER, GROUP, ORDER, and FOREACH operations on the data.		
		9.Create a Hive table for student records, insert or load sample data, and execute HiveQL queries using SELECT, WHERE, ORDER BY, GROUP BY and JOIN		

MGU-BCA (HONOURS) Syllabus

		10.Create a Hive table containing employee or student records and use **COUNT(), AVG(), MAX(), and MIN()** to obtain summary information.		

Teaching and Learning Approach	**Classroom Procedure (Mode of transaction)** • ICT enabled Lecture • Interactive sessions • Class discussions • Practical Sessions
Assessment Types	**MODE OF ASSESSMENT** **A. Continuous Comprehensive Assessment (CCA)** **CCA for Theory: 30 Marks** 1. Written tests 2. Assignments 3. Viva 4. MCQ **CCA for Practical: 15 Marks** 1. Practical assignments 2. Observation of practical skills, Viva etc.
	B. End Semester Evaluation **ESE for Theory: 70 Marks (2 Hrs)** Written Test **(70 Marks)** Part A: Very Short Answer Questions (Answer all) - (5*2=10 Marks) Part B: Short Answer Questions (5 out of 7 Questions) - (5*6=30 Marks) Part C: Essay Questions (2 out of 3 Questions) - (2*15=30 Marks)

MGU-BCA (HONOURS)

	ESE for Practical: 35 Marks (1.5 Hrs) Logic - 10 Marks Successful Compilation – 5 Marks Output - 5 Marks Viva – 10 Marks Record – 5 Marks

REFERENCES:

1. Seema Acharya, Subhasini Chellappan(2015). "Big Data Analytics". Wiley. (Module 1,2,3,4).

2. Tom White(2012). " Hadoop: The Definitive Guide"(Third Edition). O'reilly Media. (Module 3)

SUGGESTED READINGS:

1. Anand Rajaraman and Jeffrey David Ullman, "Mining of Massive Datasets", Cambridge University Press, 2012.

2. Bill Franks, "Taming the Big Data Tidal Wave: Finding Opportunities in Huge Data Streams with Advanced Analytics", John Wiley & sons, 2012.

3. Glenn J. Myatt, "Making Sense of Data", John Wiley & Sons, 2007.

4. Pete Warden, "Big Data Glossary", O'Reilly, 2011.

MAHATMA GANDHI UNIVERSITY · KOTTAYAM · विद्यया अमृतमश्नुते
MGU-BCA (HONOURS)
Syllabus

	Mahatma Gandhi University **Kottayam**
Programme	**BCA (Honours)**
Course Name	**Exploratory Data Analysis**
Type of Course	**DSE**
Course Code	**MG6DSEBCA301**
Course Level	**NA**
Course Summary	This course teaches students to examine a raw dataset before any modelling: to understand its structure, clean it, summarise it and find patterns, anomalies and relationships through statistics and visualisation. It follows Tukey's idea of letting the data suggest hypotheses rather than testing assumed ones. The course is hands-on and Python-based (Pandas, NumPy, Matplotlib, Seaborn, Plotly). Every theory module is paired with lab work on real public datasets, and the course ends with a mini project in which students produce a complete EDA report.

Semester	6	Credits			5	Total Hours
Course Details	Learning Approach	Lecture	Tutorial	Practical	Others	
		4	0	1	0	90
Pre-requisites, if any	Python programming; basic statistics and probability					

COURSE OUTCOMES (CO)

CO No.	Expected Course Outcome	Learning Domains *	PO No
1	Explain the Exploratory Data Analysis(EDA) process, data types, data quality and Python EDA tools.	U	1
2	Apply techniques to clean, transform and reshape raw data for analysis.	A	2
3	Analyse patterns and anomalies using statistical and algorithmic methods	An	1,2
4	Build effective visualisations to explore data and communicate findings	A	2,3
5	Perform end-to-end EDA on real-world data in Python and report findings.	A	2,3

Remember (K), Understand (U), Apply (A), Analyse (An), Evaluate (E), Create (C), Skill (S), Interest (I) and Appreciation (Ap)

MGU-BCA (HONOURS) Syllabus

COURSE CONTENT

Content for Classroom transactions (Units)

Module	Units	Course description	Hrs	CO No.
1		**Foundations of EDA and Data Understanding**		
	1.1	**Introduction to EDA:** Definition and goals of EDA; EDA vs confirmatory data analysis; Tukey's approach; EDA in the data science lifecycle (CRISP-DM); EDA vs reporting; role of EDA before machine learning.	4	1
	1.2	**Understanding Data:** Data, information and knowledge; structured, semi-structured and unstructured data; cross-sectional, time-series and panel data; levels of measurement (nominal, ordinal, interval, ratio); categorical vs numerical variables; data sources – files, databases, APIs, open data portals (data.gov.in, Kaggle, UCI).	4	1
	1.3	**Data Quality:** Dimensions of data quality – accuracy, completeness, consistency, timeliness, validity, uniqueness; common data quality problems.	3	1
	1.4	**Tools for EDA:** Python ecosystem – Jupyter Notebook / Google Colab, NumPy, pandas, Matplotlib, Seaborn; Series and DataFrame; reading CSV, Excel, JSON and SQL data; first look at data – head, tail, shape, dtypes, info, describe, value_counts, unique; selecting, filtering and sorting.	4	1
2		**Data Cleaning and Transformation**		
	2.1	**Missing Data:** Types of missingness – MCAR, MAR, MNAR; detecting and visualising missing values; deletion methods; imputation – mean, median, mode, forward/backward fill, interpolation, KNN imputation.	3	2
	2.2	**Data Inconsistencies:** Duplicate records; inconsistent categories and spellings; type conversion; string cleaning; parsing dates and times.	3	2
	2.3	**Outliers:** Causes of outliers; detection using z-score, IQR rule and box plots; treatment – removal, capping (winsorisation), transformation.	3	2

MGU-BCA (HONOURS) Syllabus

	2.4	**Data Transformation:** Scaling – min-max normalisation, standardisation; log, square-root and Box-Cox transformations; binning / discretisation; encoding categorical variables – label and one-hot encoding; deriving new features.	3	2
	2.5	**Combining and Reshaping Data:** merge, join and concat; groupby and aggregation; pivot tables; melt and stack/unstack; crosstab.	3	2
3		**Statistical and Algorithmic Techniques for EDA**		
	3.1	**Statistical Foundations:** Measures of central tendency and dispersion; percentiles, quartiles and five-number summary; skewness and kurtosis; normal distribution and Q-Q plots; covariance, Pearson and Spearman correlation; contingency tables and chi-square test; t-test and one-way ANOVA (applied with SciPy); correlation vs causation; Simpson's paradox.	5	3
	3.2	**Similarity, Distance and Information Measures:** Distance and similarity metrics – Euclidean, Manhattan, Minkowski, cosine, Hamming, Jaccard; edit (Levenshtein) distance and fuzzy matching for detecting near-duplicate records; entropy, information gain and mutual information for exploring feature–target relationships; decision-tree-based feature importance as an exploratory tool.	5	3
	3.3	**Algorithmic Anomaly Detection and Scalable EDA:** Distance- and density-based outliers – k-NN distance, Local Outlier Factor (LOF); Isolation Forest. EDA on large data – single-pass algorithms (Welford's online mean and variance), reservoir sampling, hashing for duplicate detection, approximate distinct counting (HyperLogLog, concept only).	5	3
		Data Visualisation and Multivariate EDA		
	4.1	**Principles of Visualisation:** Purpose of visualisation; choosing the right chart for the data; Tufte's principles and the data-ink ratio; colour and perception; misleading charts and visualisation ethics.	2	4
	4.2	**Visualisation with Matplotlib and Seaborn:** Anatomy of a Matplotlib figure; line, bar, histogram, pie, scatter and box plots; subplots; Seaborn – histplot, kdeplot, countplot, boxplot, violinplot, scatterplot, heatmap, pairplot, FacetGrid.	4	4

MGU-BCA (HONOURS) Syllabus

4	4.3	**Multivariate and Time-Series EDA:** Multivariate plots – pair plots, grouped and faceted charts, bubble charts; dimensionality reduction for exploration – PCA intuition and explained variance; time-series EDA – trend, seasonality, rolling statistics, decomposition.	4	4
	4.4	**Interactive and Automated EDA:** Interactive charts with Plotly Express; automated profiling tools – ydata-profiling, Sweetviz; limits of automated EDA.	3	4
	4.5	**Communicating Findings:** Structure of an EDA report; data storytelling; presenting insights to non-technical audiences.	2	4
5		**Practical**		
		Hands-on application of Modules 1–4 in Python (Jupyter Notebook / Google Colab) on real-world datasets. The Module 5 practical hours are delivered through these assignments, using public datasets such as Titanic, Iris, Tips, Kerala rainfall data from data.gov.in, Indian census or retail sales data. 1. Load data from CSV, Excel, JSON and a SQLite table into pandas; inspect structure, data types and summary statistics; select, filter and sort. 2. Detect and visualise missing values; compare deletion with mean, median and KNN imputation. 3. Identify and remove duplicates; clean inconsistent categories, strings and date fields. 4. Detect outliers with z-score, IQR and box plots; apply capping and log transformation. 5. Apply scaling (min-max, standardisation), binning and one-hot / label encoding. 6. Combine datasets using merge and concat; summarise with groupby, pivot_table and crosstab. 7. Compute descriptive statistics, skewness and kurtosis; perform univariate analysis with histograms, Q-Q plots and count plots. 8. Perform bivariate analysis: correlation matrix with heatmap, chi-square test, group-wise comparison with t-test / ANOVA.	30	5

MGU-BCA (HONOURS) Syllabus

		9. Implement distance and similarity measures (Euclidean, cosine, Jaccard, Levenshtein); find near-duplicate records using fuzzy matching. 10. Compute entropy and mutual information between features and a target; rank features using decision-tree importance. 11. Detect anomalies with LOF and Isolation Forest; compare the results with the IQR method from Assignment 4. 12. Build multivariate visualisations with pairplot and FacetGrid; apply PCA and plot explained variance. 13. Explore a time-series dataset: trend, rolling mean and seasonal decomposition; build interactive charts with Plotly Express and an automated profile with ydata-profiling.		
		14. **Mini project:** complete EDA on a dataset of the student's choice, combining statistical and algorithmic techniques, submitted as a notebook and a short EDA report with a presentation.		

Teaching and Learning Approach	**Classroom Procedure (Mode of transaction)** • **Classroom procedure:** lecture with live Jupyter demonstrations, guided worksheets, flipped-classroom reading on statistics topics. • **Lab:** one practical per week from Module 5, recorded in a lab notebook / GitHub repository. • **Collaborative learning:** pair programming in labs; small-group dataset challenges. • **Case-based learning:** EDA case studies from public Indian datasets (rainfall, agriculture, census, retail).

MAHATMA GANDHI UNIVERSITY KOTTAYAM
विद्यया अमृतमश्नुते
MGU BCA (HONOURS)
Syllabus

Assessment Types	**MODE OF ASSESSMENT** **A. Continuous Comprehensive Assessment (CCA)** **CCA for Theory: 30 Marks** ● Written test ● Assignments ● Seminar ● MCQ/Viva **CCA for Practical: 15 Marks** ● Practical assignments ● Observation of practical skills ● Viva etc.
	B. End Semester Evaluation (ESE) **ESE for Theory: Written Test (70 Marks, 2 Hrs)** Part A: Very Short Answer Questions. (Answer all) – (5 x 2 = 10 marks) Part B: Short answer questions (5 out of 7 Questions) – (5 x 6 = 30 marks) Part C: Essay questions (2 out of 3 Questions)- (2 x 15 = 30 marks) **ESE for Practical: (35 Marks)** 1. Project Demonstration - 15 Marks 2. Project Report - 10 Marks 3. Viva - 10 Marks

MGU-BCA (HONOURS)

References

1. Suresh Kumar Mukhiya and Usman Ahmed, *Hands-On Exploratory Data Analysis with Python*, Packt Publishing, 2020. (Modules 1–4)
2. Wes McKinney, *Python for Data Analysis*, 3rd Edition, O'Reilly Media, 2022. (Modules 1, 2)
3. Peter Bruce, Andrew Bruce and Peter Gedeck, *Practical Statistics for Data Scientists*, 2nd Edition, O'Reilly Media, 2020. (Module 3)
4. Jiawei Han, Jian Pei and Hanghang Tong, *Data Mining: Concepts and Techniques*, 4th Edition, Morgan Kaufmann, 2022. (Module 3)

Syllabus

Suggested Readings

1. John W. Tukey, *Exploratory Data Analysis*, Addison-Wesley, 1977.
2. Jure Leskovec, Anand Rajaraman and Jeffrey D. Ullman, *Mining of Massive Datasets*, 3rd Edition, Cambridge University Press, 2020. (similarity, hashing and streaming algorithms)
3. Jake VanderPlas, *Python Data Science Handbook*, 2nd Edition, O'Reilly Media, 2023.
4. Claus O. Wilke, *Fundamentals of Data Visualization*, O'Reilly Media, 2019.
5. Ronald K. Pearson, *Exploratory Data Analysis Using R*, CRC Press, 2018.
6. Hadley Wickham, Mine Çetinkaya-Rundel and Garrett Grolemund, *R for Data Science*, 2nd Edition, O'Reilly Media, 2023 (chapters on EDA and visualisation).

Web Resources

- Pandas documentation: pandas.pydata.org
- Seaborn tutorial: seaborn.pydata.org
- Plotly Express: plotly.com/python
- Open Government Data Platform India: data.gov.in
- UCI Machine Learning Repository: archive.ics.uci.edu

MGU-BCA (HONOURS)

Syllabus

Mahatma Gandhi University
Kottayam

Programme	**BCA (Honours)**					
Course Name	**Deep Learning for Computer Vision**					
Type of Course	**DSE**					
Course Code	**MG6DSEBCA302**					
Course Level	**NA**					
Course Summary	This course introduces students to the fundamentals of computer vision, object detection, and image segmentation. Introduces deep learning techniques. Students will also gain a basic understanding of attention mechanisms, transformers, deep generative models, and recent developments in computer vision. The practical component provides hands-on experience in implementing computer vision techniques using Python and deep learning frameworks.					
Semester	6	Credits			5	Total Hours
Course Details	Learning Approach	Lecture	Tutorial	Practical	Others	
		4	0	1	0	90
Pre-requisites, if any	Programming Knowledge in Python, Basic understanding of Image processing, Artificial Intelligence and Machine Learning.					

MGU-BCA (HONOURS)

COURSE OUTCOMES (CO)

CO No.	Expected Course Outcome	Learning Domains *	PO No
1	Explain the basics of image processing and computer vision.	U	1,2
2	Analyse deep learning for computer vision.	An	1,2
3	Analyse segmentation and object detection architectures for computer vision tasks.	An	2,3
4	Examine advanced deep learning architectures, emerging trends, challenges and opportunities in computer vision.	An	1,2
5	Implement deep learning techniques for computer vision applications using Python.	A	2,3

**Remember (K), Understand (U), Apply (A), Analyse (An), Evaluate (E), Create (C), Skill (S), Interest (I) and Appreciation (Ap)*

COURSE CONTENT

Content for Classroom transaction (Units)

Module	Units	Course description	Hrs	CO No.
1	**Introduction to Image Processing and Computer Vision (15 hrs)**			
	1.1	**Introduction to Computer Vision:** Fundamentals and scope of computer vision, Everyday applications. Basic steps involved in designing a simple computer vision system.	2	1
	1.2	**Image Formation**: Fundamentals of image formation in computer vision. Light, scene geometry, and image capture. Basic imaging hardware, image sensors, and illumination sources. Histograms, Introduction to convolution and Kernels, image filtering using Kernels.	3	1
	1.3	**Edges and Features:** Fundamentals of edge detection and feature extraction in digital images; introduction to Scale-Invariant Feature Transform (SIFT) and Oriented FAST and Rotated BRIEF (ORB), including their basic principles, keypoint detection, feature description, and applications in image matching and object recognition. Limitations of handcrafted feature extraction techniques- dependence on manually designed descriptors, sensitivity to variations in scale, rotation, illumination, and complex image patterns.	4	1
	1.4	**Introduction to Image Transformation:** Fundamentals and properties of orthogonal, Euclidean, affine, and projective transformations; their mathematical representations and applications in image processing and computer vision. Introduction to the Fourier Transform-mathematical properties, frequency-domain representation;	6	1

MGU-BCA (HONOURS) Syllabus

		applications of Fourier Transform- in image enhancement, filtering, and image analysis.		
2	**Deep Learning for Computer Vision (15 hrs)**			
	2.1	**Introduction to Convolutional Neural Networks (CNN):** Basic concepts - Convolution, pooling and fully connected layers. Loss Functions, Forward and Backward Propagation	3	2
	2.2	**Training a CNN:** Fundamentals of CNN training- preparation and organization of datasets into training, validation, and testing sets; Understanding the training process-forward propagation, loss computation, backpropagation; weight optimization; Introduction to loss functions- categorical cross-entropy and binary cross-entropy; Analysis of training and validation performance- Understanding overfitting and underfitting in CNN models, their causes and impact on model generalization.	5	2
	2.3	**Popular CNN Models:** Introduction to VGG and ResNet- characteristics, architecture and working principles; **Transfer Learning and its Applications**: Pre-trained models. Working with small datasets. Using a pre-trained model on a new dataset. Current Trends and Future of Deep Learning	4	2
	2.4	**Evaluation criteria for CNN models:** Evaluation metrics- accuracy, precision, recall -mathematical formulations, interpretations; Detailed study of the confusion matrix- structure, construction, interpretation, and application in analysing correct and incorrect classifications.	3	2
3	**Segmentation and Object Detection (15 hrs)**			
	3.1	**Object Detection Basics:** Introduction to object detection in computer vision- objectives, fundamental principles, applications; Classification vs detection; Bounding boxes; Introduction to Intersection over Union (IoU)-	3	3

MGU-BCA (HONOURS)

		mathematical formulation, interpretation, and significance in object localization and detection accuracy.		
	3.2	**Detection Methods:** Region based object detection approaches- architecture and working principles of two stage detectors- R-CNN and Faster R-CNN; improvements of Faster R-CNN over R-CNN . Introduction to single-stage object detection approaches- YOLO architecture and basic working principles.	4	3
	3.3	**Image Segmentation Methods:** Introduction- Semantic vs instance segmentation.; Global Thresholding method; Region Growing; Edge Based approaches to segmentation, Introduction to Graph-Cut algorithms for image segmentation.	3	3
	3.4	**Deep learning based segmentation approaches:** Fully Convolutional Networks (FCN)- .Architecture and fundamental principles Role of FCN in object detection; Evaluation strategies and applications; U-Net- architecture and its fundamental principles in deep learning-based computer vision applications; skip connections.	5	3
4	**Advanced Deep Learning Architectures for computer vision (15 hrs)**			
	4.1	**Attention Models:** Introduction to attention mechanisms, motivation and basic principles of attention; Self-attention mechanism and its role in feature representation; Applications of attention mechanisms in deep learning.	3	4
	4.2	**Vision Transformer:** Transformer based architectures for computer vision-fundamental principles; Image patch extraction-division of input images into fixed size patches, flattening of image patches, linear projection, and conversion into patch embeddings; Vision Transformer (ViT)- Application in image classification, ViT vs. CNN.	5	4
	4.3	**Generative Models:** Generative models- fundamental principles and objectives, Variational Autoencoders	5	4

MGU-BCA (HONOURS) Syllabus

		(VAEs), Generative Adversarial Networks (GANs), and diffusion-based models; Generative vs. discriminative models; **Applications of Generative Models:** Image generation, inpainting, super-resolution. Deepfakes and ethical concerns.		
	4.4	**Trends in Computer Vision:** multimodal learning approaches; self-supervised learning; few-shot and zero-shot learning; explainable and trustworthy computer vision; challenges and opportunities.	2	4
5	**Lab Practice** Implement the lab assignments using Python and public data sets such as CIFAR-10, MS COCO, PASCAL VOC, Open Images Dataset			
	5.1	1. Image Transformations: Orthogonal, Euclidean, affine, and projective transformations 2. Basic edge detection methods. 3. Discrete Fourier Transform (DFT) to an image and visualize its frequency-domain representation 4. SIFT and ORB feature detectors and descriptors to identify and match key points between images 5. Image segmentation: thresholding, Region Growing, Edge Based approaches to segmentation, Graph-Cut. 6. Basic object detection methods 7. Simple convolutional neural network to understand convolution, pooling, and fully connected layers. 8. CNN model to classify images from a standard image dataset. 9. Train and test a CNN using separate datasets and analyze training and validation loss and accuracy. 10. Use pre-trained CNN models to classify images from a new dataset and print Performance evaluation metrics such as confusion matrix.	30	5

MAHATMA GANDHI UNIVERSITY, KOTTAYAM
MGU-BCA (HONOURS)
Syllabus

Teaching and Learning Approach	**Classroom Procedure /Mode of transaction** • Lecturing • Demonstrations • Collaborative learning • Dataset-based experimentation
Assessment Types	**MODE OF ASSESSMENT** ● **Continuous Comprehensive Assessment (CCA)** **CCA for Theory: 30 Marks** 1. Written test 2. Assignments 3. Viva **CCA for Practical: 15 Marks** 1. Practical assignments 2. Lab Record 3. Observation of practical skills 4. Viva
	● **End Semester Evaluation** **ESE for Theory: Written Test(70 Marks, 2 Hours)** Part A: Very Short Answer Questions. (Answer all) – (5 x 2 = 10 marks) Part B: Short answer questions (5 out of 7 Questions) – (5 x 6 = 30 marks). Part C: Essay questions (2 out of 3 Questions)- (2 x 15 = 30 marks) **ESE for Practical: (35 Marks, 1.5 Hrs)** 1. Understanding of the problem and selection of an appropriate computer vision approach - 10 Marks 2. Output : Result analysis and interpretation - 10 Marks 3. Viva - 10 Marks 4. Record - 5 Marks

Syllabus

REFERENCES

1. Rafael C. Gonzalez and Richard E. Woods, Digital Image Processing, 4th Edition, Pearson, 2018. (Module 1 and 3)
2. Rajalingappaa Shanmugamani, *Deep Learning for Computer Vision*, Packt Publishing, 2018. (Modules 2–4)
3. Mohamed Elgendy, *Deep Learning for Vision Systems*, Manning Publications, 2020. (Modules 2–4)
4. Richard Szeliski, *Computer Vision: Algorithms and Applications*, 2nd Edition, Springer, 2022. (Modules 1 and 3)
5. Ian Goodfellow, Yoshua Bengio and Aaron Courville, *Deep Learning*, MIT Press, 2016. (Module 4)

SUGGESTED READINGS

1. François Chollet, *Deep Learning with Python*, 2nd Edition, Manning Publications, 2021.
2. Aurélien Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*, 3rd Edition, O'Reilly Media, 2022.
3. Adrian Rosebrock, *Deep Learning for Computer Vision with Python*, PyImageSearch, 2017.
4. Gary Bradski and Adrian Kaehler, *Learning OpenCV 3: Computer Vision in C++ with the OpenCV Library*, O'Reilly Media, 2016.
5. David Foster, *Generative Deep Learning*, 2nd Edition, O'Reilly Media, 2023.
6. Aston Zhang, Zachary C. Lipton, Mu Li and Alexander J. Smola, *Dive into Deep Learning*, Cambridge University Press, 2023.
7. Simon J. D. Prince, *Understanding Deep Learning*, MIT Press, 2023.

MAHATMA GANDHI UNIVERSITY · KOTTAYAM · MGU-BCA (HONOURS) Syllabus

Programme	**BCA (Honours)**					
Course Name	**Predictive Analytics**					
Type of Course	**DSE**					
Course Code	**MG6DSEBCA303**					
Course Level	**NA**					
Course Summary	The course imparts practical knowledge of data analytics concepts, methods and their implementation in Python, organised around the CRISP-DM lifecycle—from business framing, data exploration, data reduction and performance evaluation to regression, ensemble (XGBoost) and forecasting models, and model deployment.					
Semester	6	**Credits**	5			**Total Hours**
Course Details	**Learning Approach**	Lecture	Tutorial	Practical	Others	90
		4	0	1	0	
Pre-requisites, if any	Computer fundamentals, Python programming and basic statistics.					

COURSE OUTCOMES (CO)

CO No.	Expected Course Outcome	Learning Domains *	PO No
1	Explain data analytics concepts, the CRISP-DM lifecycle, business framing and the principles of data preparation, partitioning and leakage prevention.	U	1,2,3
2	Analyse data through visualisation, data reduction and business-oriented performance evaluation.	An	1,2,3
3	Apply multiple linear regression, variable selection, regularisation and logistic regression to business problems and interpret the results.	A	1,2,3
4	Evaluate ensemble models, including XGBoost, and time-series forecasting methods for prediction.	E	1,2,3
5	Develop and communicate an end-to-end data analytics workflow in Python, including deployment and monitoring considerations.	C	2,3
***Remember (K), Understand (U), Apply (A), Analyse (An), Evaluate (E), Create (C), Skill (S), Interest (I) and Appreciation (Ap)**			

MGU BCA (HONOURS) Syllabus

COURSE CONTENT

Content for Classroom transaction (Units)

Module	Units	Course description	Hrs	CO No.
1	1.1	Introduction to data analytics: business analytics, data mining, big data and data science—definitions and related terms; types of analytics—descriptive, diagnostic, predictive and prescriptive; core ideas—prediction, classification, association rules and recommendation systems, data reduction, data exploration; supervised and unsupervised learning (T1 Ch. 1, 2.1–2.2).	3	1
	1.2	Data analytics lifecycle—CRISP-DM: business understanding, data understanding, data preparation, modelling, evaluation and deployment; iterative nature of real projects; business framing—translating stakeholder requests into testable predictive hypotheses, prediction versus explanation, cost-sensitive thinking (false positives versus false negatives) (R1; R2 Ch. 1–2; T1 Ch. 2.3).	3	1
	1.3	Preliminary steps: organisation of datasets; loading and inspecting data in Python; sampling from a database; oversampling rare events; preprocessing and cleaning the data; missing-value mechanisms—MCAR, MAR and MNAR (T1 Ch. 2.4).	3	1
	1.4	Data partitioning and overfitting: predictive power and overfitting; training, validation and test partitions; stratified and temporal splits; never preprocess before splitting; data leakage; bias–variance trade-off and learning curves (T1 Ch. 2.5; T2 Ch. 2; T4 Ch. 4).	3	1
	1.5	Building a predictive model in Python: leakage-free preprocessing with Pipeline and ColumnTransformer; automating data analytics solutions; ethical practice in data analytics (T1 Ch. 2.6–2.9; R6).	3	1

MGU BCA (HONOURS) Syllabus

2	2.1	Data visualisation: basic charts—bar charts, line graphs and scatter plots; distribution plots—boxplots and histograms; heatmaps for correlations and missing values (T1 Ch. 3.1–3.3).	3	2
	2.2	Multidimensional visualisation: adding variables through colour, size, shape and multiple panels; rescaling, aggregation, zooming and filtering; parallel coordinates and interactive visualisation; correlation heatmaps as hypothesis generators (T1 Ch. 3.4–3.6).	3	2
	2.3	Data reduction: curse of dimensionality; data summaries; correlation analysis; reducing the number of categories in categorical variables; converting a categorical variable to a numerical variable; dimension reduction using regression models (T1 Ch. 4.1–4.7, 4.9).	3	2
	2.4	Evaluating numeric predictions: naive benchmark; prediction accuracy measures—RMSE, MAE, MAPE and R^2; comparing training and validation performance; residual diagnostics; business cost functions for asymmetric over- and under-prediction costs (T1 Ch. 5.1–5.2; T3 Ch. 5).	3	2
	2.5	Evaluating classification for decisions: confusion matrix and ROC-AUC (recap); propensities and the cutoff—why 0.5 is rarely optimal; lift charts and ranking performance; precision–recall curves; asymmetric misclassification costs and cutoff selection from a cost matrix; oversampling (T1 Ch. 5.3–5.5; T3 Ch. 11).	3	2
3	3.1	Multiple linear regression: explanatory versus predictive modelling; matrix formulation $y = X\beta + \varepsilon$; ordinary least squares and the normal equations; estimating the regression equation and prediction using statsmodels (T1 Ch. 6.1–6.3; T2 Ch. 3).	3	3
	3.2	Business interpretation of regression: coefficients as effect sizes; confidence intervals; statistical versus practical significance; quantifying the value of an intervention (T2 Ch. 3; R2 Ch. 7).	3	3
	3.3	Variable selection in linear regression: reducing the number of	3	3

MGU-BCA (HONOURS)

		predictors; exhaustive search; forward selection, backward elimination and stepwise regression (T1 Ch. 6.4).		
	3.4	Regularisation: ridge (L2), lasso (L1) and elastic net; regularisation paths; cross-validated selection of λ (T2 Ch. 6; T3 Ch. 6).	3	3
	3.5	Logistic regression: the logistic model; maximum likelihood estimation and log loss; interpreting results in terms of odds; evaluating classification performance; variable selection; logistic regression for more than two classes (T1 Ch. 10; T2 Ch. 4).	3	3
4	4.1	Ensembles and gradient boosting: bagging reduces variance and boosting reduces bias; simple averaging; random forests and out-of-bag error (recap); gradient boosting as sequential residual fitting; XGBoost—regularised objective, learning rate, subsampling, native missing-value handling and early stopping (T1 Ch. 13; T2 Ch. 8; T4 Ch. 7; R7).	3	4
	4.2	Interpreting models and uplift modelling: impurity-based (Gini) versus permutation feature importance; bias towards high-cardinality features; partial dependence plots; uplift (persuasion) modelling (T1 Ch. 13; R6).	3	4
	4.3	Time-series analysis: time-series components; data partitioning and performance evaluation for time series; naive and seasonal naive benchmarks; time-series decomposition—additive, multiplicative and STL (T1 Ch. 16; R3 Ch. 3, 5).	3	4
	4.4	Forecasting methods: regression-based forecasting with trend and seasonality; autocorrelation and ARIMA models—stationarity and differencing; smoothing methods—moving average, simple exponential smoothing and Holt–Winters (T1 Ch. 17–18; R3 Ch. 7–9).	3	4
	4.5	Deployment and communication: saving models with joblib; batch versus real-time scoring; reproducibility; data drift and concept drift—population stability index and retraining;	3	5

MGU-BCA (HONOURS)

		structuring results—context, approach, findings, limitations and recommendations; fairness and privacy (T4 Ch. 2; R5; R6).		
5	5.1	**Lab Practice: Implement using Python:** 1. Import a business dataset into pandas; inspect its shape, data types and summary statistics; detect and treat missing values and outliers, recording the cleaning decisions taken. 2. Create stratified and temporal training, validation and test partitions with a fixed random seed; build a leakage-free Pipeline with ColumnTransformer and correct a deliberate data-leakage scenario. 3. Produce bar charts, line graphs, scatter plots, boxplots and histograms for a business dataset using matplotlib. 4. Build correlation and missing-value heatmaps using seaborn and a multidimensional scatter plot using colour, size, shape and multiple panels. 5. Reduce a high-cardinality categorical variable by consolidating rare categories and convert categorical variables to numerical form. 6. Compute a naive benchmark and prediction accuracy measures (RMSE, MAE, MAPE) for a numerical outcome and evaluate it with a custom asymmetric cost function. 7. Plot cumulative and decile lift charts; vary the classification cutoff, plot performance against the cutoff and select a cutoff from a cost matrix. 8. Fit a multiple linear regression model (e.g., customer lifetime value) using statsmodels and interpret the coefficients for a marketing manager. 9. Perform variable selection using forward, backward and stepwise methods and compare validation performance. 10. Fit ridge and lasso models on a dataset with many predictors; plot the regularisation path and select λ by cross-validation. 11. Fit a logistic regression model for customer churn and interpret the results in terms of odds. 12. Build random forest (with out-of-bag error) and XGBoost (with early stopping) models and an averaging ensemble; compare them with logistic regression using ROC-AUC and log loss. 13. Compute permutation feature importance and partial dependence plots and write stakeholder-ready interpretations. 14. Decompose a retail sales series; forecast with seasonal naive, regression with trend and seasonality, and Holt–Winters models; compare them on a validation period. 15. Capstone presentation (project developed outside lab hours; choose one: customer churn, credit default or demand forecasting): build the end-to-end workflow, save the model with joblib and batch-score new data; submit a reproducible notebook with requirements.txt and a short business memo.	30	5

MGU-BCA (HONOURS)

Teaching and Learning Approach	**Classroom Procedure /Mode of transaction** • Lecturing • Demonstrations • Collaborative learning • Self-directed learning • ICT enabled lecture
Assessment Types	**MODE OF ASSESSMENT** **A. Continuous Comprehensive Assessment (CCA)** **CCA for Theory: 30 Marks** 1. Written test 2. Assignments 3. Viva 4. MCQ etc. **CCA for Practical: 15 Marks** 1. Practical assignments 2. Lab Record 3. Observation of practical skills 4. Viva
	B. End Semester Evaluation **ESE for Theory: Written Test(70 Marks, 2 Hours)** Part A: Very Short Answer Questions. (Answer all) – (5 x 2= 10 marks) Part B: Short answerquestions (5outof 7 Questions) – (5 x 6 = 30 marks). Part C: Essay questions (2 out of 3 Questions)- (2x15=30marks) **ESE for Practical: (35 Marks, 1.5 Hrs)** 1. Coding - 10 Marks 2. Output - 10 Marks 3. Viva - 10 Marks 4. Record - 5 Marks

MAHATMA GANDHI UNIVERSITY KOTTAYAM
MGU-BCA (HONOURS)
Syllabus

REFERENCES (TEXTBOOKS)

T1. Shmueli, G., Bruce, P. C., Gedeck, P., & Patel, N. R. (2020). Data Mining for Business Analytics: Concepts, Techniques and Applications in Python. Wiley India Pvt. Ltd.

T2. James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). An Introduction to Statistical Learning (2nd ed.). Springer.

T3. Kuhn, M., & Johnson, K. (2013). Applied Predictive Modeling. Springer.

T4. Géron, A. (2022). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow (3rd ed.). O'Reilly Media.

SUGGESTED READINGS:

R1. Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., & Wirth, R. (2000). CRISP-DM 1.0: Step-by-Step Data Mining Guide. CRISP-DM Consortium.

R2. Provost, F., & Fawcett, T. (2013). Data Science for Business. O'Reilly Media.

R3. Hyndman, R. J., & Athanasopoulos, G. (2021). Forecasting: Principles and Practice (3rd ed.). OTexts.

R4. Bruce, P., Bruce, A., & Gedeck, P. (2020). Practical Statistics for Data Scientists (2nd ed.). O'Reilly Media.

R5. Barocas, S., Hardt, M., & Narayanan, A. (2023). Fairness and Machine Learning. fairmlbook.org.

R6. scikit-learn User Guide (current stable release).

R7. XGBoost Documentation—Parameter Tuning and Early Stopping.

R8. McKinney, W. (2022). Python for Data Analysis (3rd ed.). O'Reilly Media.

R9. Bari, A., Chaouchi, M., & Jung, T. (2014). Predictive Analytics For Dummies. John Wiley & Sons, Hoboken, New Jersey.

MAHATMA GANDHI UNIVERSITY KOTTAYAM
विद्यया अमृतमश्नुते
MGU-BCA (HONOURS)
Syllabus

	Mahatma Gandhi University **Kottayam**
Programme	**BCA (Honours)**
Course Name	**DevOps Fundamentals with Docker and Jenkins**
Type of Course	**DSE**
Course Code	**MG6DSEBCA304**
Course Level	**NA**
Course Summary	This course introduces the principles and practices of DevOps, including continuous integration and continuous delivery. Students learn essential Linux shell commands, basic shell scripting and version control with Git. The course covers containerization with Docker, including images, Dockerfiles, volumes, networking and Docker Compose, followed by build automation and CI/CD pipelines using Jenkins. Through hands-on lab work and a semester project, students build, containerize and deploy a simple web application using Git, Docker and Jenkins.

Semester	6	Credits		5		Total Hours
Course Details	Learning Approach	Lecture	Tutorial	Practical	Others	90
		4	0	1	0	
Pre-requisites if any	Basic knowledge of Web Technologies.					

COURSE OUTCOMES (CO)

CO No	Expected Course Outcomes	Learning Domains*	PO No
1	Apply DevOps and CI/CD concepts using Linux shell commands, simple shell scripts and Git for version control.	A	1,2
2	Apply Docker concepts to install Docker, manage images and registries, and run and manage containers.	A	2
3	Apply Dockerfiles, volumes, networking and Docker Compose to build, containerize and run multi-container web applications.	A	2

MGU-BCA (HONOURS) Syllabus

4	Apply Jenkins to create build jobs and pipelines, and automate the deployment of Docker applications using Docker Compose.	A	2
5	Apply Linux, Git, Docker and Jenkins to develop, containerize and deploy a web application through a CI/CD workflow.	A	2,3
***Remember (K), Understand (U), Apply (A), Analyse (An), Evaluate (E), Create (C), Skill (S), Interest (I) and Appreciation (Ap)**			

COURSE CONTENT

Content for Classroom transaction (Units)

Module	Units	Course description	Hrs	CO No.
1	1.1	**Introduction to DevOps:** Need for DevOps, DevOps culture and principles. Continuous Integration: automated builds, automated tests, CI servers, trunk-based development. Continuous Delivery vs Continuous Deployment.	4	1
	1.2	**Linux Shell Basics:** the shell and terminal, navigating the file system, exploring the system, manipulating files and directories, basic commands and file permissions.	3	1
	1.3	**Shell Scripting Basics:** Writing and executing shell script, variables, operators, basic conditional statements and loops.	3	1
	1.4	**Version Control with Git:** version control concepts; git init, clone, add, commit, status, log; branching and merging; working with remote repositories (push, pull) on a Git hosting platform.	5	1
	2.1	**Need for Containers:** Docker introduction, virtual machines vs containers, what a container is.	3	2
	2.2	**Docker Overview:** the Docker platform, Docker architecture (client, daemon, registries), Docker objects	3	2

MGU-BCA (HONOURS) Syllabus

2		(images, containers). Installing Docker Engine on Linux and post-installation steps.		
	2.3	**Images and Registries:** what an image is, image layers, image naming and tags, pulling, listing and removing images, image history. Public and private registries, repositories.	4	2
	2.4	**Running Containers:** container lifecycle and commands (run, ps, stop, start, restart, rm); interactive vs detached mode; logs, exec and inspect; publishing and exposing ports.	5	2
3	3.1	**Writing a Dockerfile:** FROM, WORKDIR, COPY, RUN, ENV, EXPOSE, USER, CMD and ENTRYPOINT. Containerizing a simple web application.	4	3
	3.2	**Building Images:** build, tag and publish an image; using the build cache and ordering instructions; Introduction to multi-stage builds.	4	3
	3.3	**Data and Networking:** persisting container data with volumes, sharing local files with bind mounts, Introduction to Docker networking - types.	3	3
	3.4	**Docker Compose:** need for Compose; compose.yaml structure (services, image/build, ports, environment, volumes, networks), running a multi-container application.	4	3
4	4.1	**Introduction to Jenkins:** role of a CI server, Jenkins features. Jenkins terminology: controller, agent, node, executor, project/job, build, workspace, plugin, pipeline, stage, step.	4	4
	4.2	**Setting up Jenkins:** Managing plugins, using credentials. Creating a freestyle project; build triggers (poll SCM, webhooks)	4	4

MGU-BCA (HONOURS) Syllabus

	4.3	**Jenkins Pipeline:** Pipeline introduction, Need for pipeline, concepts, syntax overview.	4	4
	4.4	Overview of using Docker with Jenkins. Deploying an application with Docker Compose.	3	4
	Lab Practise			
5		1. **Getting Started with Linux for DevOps** - Create directories and files; copy, move and delete files; use pwd, ls, cd, mkdir, touch, cp, mv, rm and cat; create and execute a simple shell script. 2. **Creating and Managing a Git Repository** - Install Git; create a folder for a simple HTML project; initialise a Git repository; add files; commit changes; view commit history using *git log*. 3. **Running a Web Application Using Docker** - Install Docker; download an nginx image; create and run an Nginx container; map host port 8080 to container port 80; access the web page from a browser; stop and remove the container. 4. **Creating a Docker Image Using a Dockerfile** - Create a simple HTML page; write a Dockerfile; build an image; run the image as a container; access the application through a browser. 5. **Docker Application with Persistent Data** - Run a simple application using a Docker volume; create, inspect and remove a volume; demonstrate that application data remains available after removing and recreating the container. 6. **Multi-Container Application with a Database Using Docker** - Run a MySQL database and the Adminer web interface in separate containers. Create a custom Docker network to enable	30	5

MGU-BCA (HONOURS)

		communication between them, configure both containers using environment variables, create a table and insert records through the browser, and verify from another container on the same network that the data can be retrieved using the database container's name. 7. **Running a Multi-Container Application Using Docker Compose** - Write a compose.yaml that runs a MySQL database and the Adminer web interface as two services. Start the application, access Adminer from a browser, create a table and insert records. Stop and restart the application with docker compose commands and verify that the data persists through a named volume. 8. **Creating a Simple Jenkins Build Job** - Install Jenkins; create a Jenkins Freestyle project; connect it to a GitHub repository; configure Jenkins to execute a simple shell command; run the job and examine the console output. 9. **Creating a Jenkins Pipeline Using Jenkinsfile** - Add a Jenkinsfile to a simple project; create Build, Test and Finish stages; execute the pipeline; view the output of each stage; modify the project and run the pipeline again. 10. **Build and Deploy a Docker Application Using Jenkins -** Create a simple web application with a Dockerfile and a compose.yaml; store it in GitHub; create a Jenkins pipeline that clones the project, builds it and deploys it using docker compose; access the application through a browser; modify the application, push the change and run the pipeline again to see the update.		

MGU BCA (HONOURS) Syllabus

		Students should develop and containerize a small web application using Git, Docker and Jenkins by the end of the semester. The application should be maintained in a Git/GitHub repository and deployed as a Docker container. Evaluation of the completed application, Docker configuration and CI/CD workflow will be done at the time of End Semester Examination (ESE) for practical.		

Teaching and Learning Approach	**Classroom Procedure (Mode of transaction)** • Lecture • Demonstration • Practical sessions
Assessment Types	**Mode of Assessment** **A. Continuous Comprehensive Assessment (CCA)** **CCA for Theory : 30 Marks** ● Written Exams ● Oral Presentations ● Assignments **CCA for Practical : 15 Marks** ● Practical assignments ● Observation of practical skills ● Viva etc.
	B. End Semester Evaluation **ESE for Theory: Written Test (70 Marks, 2 Hours)** Part A: Very Short Answer Questions (Answer all) - (5*2=10 Marks) Part B: Short Answer Questions (5 out of 7 Questions) - (5*6=30 Marks) Part C: Essay Questions (2 out of 3 Questions) - (2*15=30 Marks) **ESE for Practical: 35 Marks** Demonstration - 15 Marks Project Report - 10 Marks Viva - 10 Marks

MGU BCA (HONOURS) Syllabus

References

1. Marco Tulio Valente, *Software Engineering: A Modern Approach*, Chapter 10: DevOps. Available at: https://softengbook.org (Module 1)

2. William E. Shotts, *The Linux Command Line*, Seventh Internet Edition. Available at: https://linuxcommand.org/tlcl.php (Module 1)
3. Scott Chacon and Ben Straub, *Pro Git*, 2nd Edition, Apress, 2014. Available at: https://git-scm.com/book/en/v2 (Module 1)
4. Docker Inc., *Docker Documentation: Get Started (Docker Concepts) and Docker Engine Manuals*. Available at: https://docs.docker.com (Modules 2, 3)
5. The Jenkins Project, *Jenkins User Handbook*. Available at: https://www.jenkins.io/doc/book/ (Module 4)

Suggested Readings

1. Bobby Iliev, *Introduction to Docker* (eBook). Available at: https://github.com/bobbyiliev/introduction-to-docker-ebook (Modules 2, 3)
2. The Jenkins Project, *Jenkins Tutorials: Build a Node.js and React App with npm*. Available at: https://www.jenkins.io/doc/tutorials/ (Modules 4, 5)
3. Docker Inc., *Docker Language-Specific Guides*. Available at: https://docs.docker.com/guides/ (Modules 3, 5)

MAHATMA GANDHI UNIVERSITY KOTTAYAM
विद्यया अमृतमश्नुते
MGU-BCA (HONOURS)
Syllabus

Mahatma Gandhi University
Kottayam

Programme	**BCA (Honours)**					
Course Name	**Web Security and Penetration Testing with Kali**					
Type of Course	**DSE**					
Course Code	**MG6DSEBCA305**					
Course Level	**NA**					
Course Summary	This course introduces the fundamentals of web application security, ethical hacking, and authorized penetration testing using Kali Linux. Students examine web technologies, reconnaissance techniques, major vulnerabilities, secure coding practices, authentication and access controls, and evidence-based vulnerability reporting. Practical sessions use locally hosted or expressly authorized training applications, linking vulnerability verification with appropriate defensive countermeasures.					
Semester	6	Credits			5	Total Hours
Course Details	Learning Approach	Lecture	Tutorial	Practical	Others	
		4	0	1	0	90
Pre-requisites, if any	Basic knowledge of Computer Networks, Web Technologies (HTML, JavaScript), Database Management Systems (DBMS), and Operating Systems / introductory Linux					

MGU-BCA (HONOURS)

COURSE OUTCOMES (CO)

Syllabus

CO No.	Expected Course Outcome	Learning Domains *	PO No
Upon completion of this course, the students will be able to:			
1	Explain web application security principles, web architecture, ethical hacking, and essential Kali Linux operations.	U	1,2
2	Apply Linux utilities and security tools to conduct authorized web application reconnaissance and basic assessments.	A	2
3	Identify common web vulnerabilities, explain their impact, and recommend suitable preventive controls.	A	2,6,8
4	Evaluate authentication, session management, access controls, API security, and secure configurations.	E	1,2,6

5	Carry out a basic supervised penetration test in a laboratory setting and produce an evidence-based security report.	E	1,2,4
Remember (K), Understand (U), Apply (A), Analyse (An), Evaluate (E), Create (C), Skill (S), Interest (I) and Appreciation (Ap)			

COURSE CONTENT

Content for Classroom transaction (Units)

Module	Units	Course description	Hrs	CO No.
1	**MODULE I — Web Security Foundations & Introduction to Kali Linux**			
	1.1	**Introduction to Web Application Security:** Web security fundamentals; threats, vulnerabilities, CIA triad, OWASP risk awareness; ethical hacking, penetration-testing phases, authorized scope, consent and rules of engagement. Reference: R1: Ch. 1–2; TB4: Ch. 1–2	4	1
	1.2	**Web Application Technologies:** Client–server architecture; IP and DNS basics; HTTP/HTTPS requests and responses, methods, headers, status codes and cookies; browsers, web servers, and static versus dynamic content. Reference: R1: Ch. 2–4; TB4: Ch. 3	4	1
	1.3	**Introduction to Kali Linux:** Kali Linux in an isolated virtual machine; Linux filesystem and shell; basic navigation and file operations; file permissions; package management; basic network-inspection commands. Reference: R2: Introduction; Ch. 1, 3–5	5	1
	1.4	**Introduction to Web Application Assessment:** Authorized reconnaissance, attack-surface awareness, vulnerability identification, responsible security testing and simple evidence capture.	2	1

MGU-BCA (HONOURS) Syllabus

		Reference: R3: Ch. 2, 9; TB4: Ch. 4, 21		
2	**MODULE II — Kali Linux Utilities and Web Application Reconnaissance**			
	2.1	**Kali Linux System and Networking Utilities:** User and file permissions; basic process management; network interfaces, DNS queries, system and service inspection; software installation and updates. Reference: R2: Ch. 1, 3–6	**3**	2
	2.2	**Web Application Reconnaissance:** Information gathering and application mapping; introductory host discovery and port scanning with Nmap; service identification; web technology identification with WhatWeb and Nikto Reference: R3: Ch. 2, 4, 6; TB4: Ch. 4	**4**	2
	2.3	**Web Application Testing Tools:** Browser Developer Tools and cURL; introduction to interception proxies; configuring Burp Suite Community or OWASP ZAP; inspecting HTTP requests, responses, cookies, parameters and API endpoints. Reference: R3: Ch. 3, 5; TB4: Ch. 4, 20	**5**	2
	2.4	**Introduction to Security Scripting:** Basic Bash variables, loops and simple scripts; introductory Python scripts for log inspection or safe automation. Reference: R2: Ch. 8, 17	**3**	2
3	**MODULE III — Web Application Vulnerabilities and Security Threats**			
	3.1	**Introduction to Hacking Web Applications:** Security-testing mindset, reconnaissance findings and application attack-surface concepts; legal and safe verification. Reference: R3: Ch. 9	**4**	3

MGU-BCA (HONOURS) Syllabus

		Cross-Site Scripting (XSS): Stored, reflected and DOM-based XSS; user and browser impact; safe demonstrations; context-aware output encoding and basic Content Security Policy. Reference: R3: Ch. 10, 22; R1: Ch. 7; R4: Ch. 12		
	3.2	**Cross-Site Request Forgery (CSRF):** Forgery of state-changing requests, associated risks, CSRF tokens, SameSite cookies and preventive controls. Reference: R3: Ch. 11, 23; R1: Ch. 8; R4: Ch. 13 **XML External Entity (XXE):**XML parser fundamentals, external entity vulnerabilities and impact; safe XML parser configuration. Reference: R3: Ch. 12, 24; R1: Ch. 15; R4: Ch. 10	**4**	3
	3.3	**Injection Vulnerabilities:** SQL injection, basic command and code injection concepts, risks of untrusted input, controlled vulnerability identification, parameterized queries and input-validation principles. Reference: R3: Ch. 13, 25; R1: Ch. 6; R4: Ch. 9–10	**4**	3
	3.4	**Denial of Service (DoS):** Application availability, resource exhaustion, introductory DoS/DDoS concepts and defensive measures (awareness only; no disruptive live testing). Reference: R3: Ch. 14, 26; R1: Ch. 17 **Third-Party Dependency Risks:** Risks from outdated libraries, exposed component versions and vulnerable packages; updates, basic dependency checking and secure integration. Reference: R3: Ch. 15, 27; R1: Ch. 14	**3**	3

MGU-BCA (HONOURS) Syllabus

4	**MODULE IV — Authentication, Defensive Security and Penetration Testing**			
	4.1	**Authentication Security:** Authentication mechanisms; password weaknesses, credential stuffing concepts, secure password storage, MFA, secure login and logout. Reference: R1: Ch. 9; R4: Ch. 6; R3: Ch. 18	**3**	4
	4.2	**Session Management and Access Control:** Sessions and cookies, secure cookie flags, session hijacking and fixation; role-based authorization, IDOR and broken object-level authorization (BOLA). Reference: R1: Ch. 10–11; R4: Ch. 7–8; R3: Ch. 18	**3**	4
	4.3	**Web API Security:** REST APIs and endpoints; authentication and authorization; disclosure of sensitive information; introductory API verification. Reference: R3: Ch. 3, 5, 18	**2**	
	4.4	**Secure Web Application Development:** Input validation, output encoding, prepared statements, CSRF defenses, HTTPS/TLS, secure configuration, dependency updates and remediation testing. Reference: R1: Ch. 6–8, 13–14; R3: Ch. 17, 19, 22–27; R4: Ch. 2, 19	**4**	4
	4.5	**Vulnerability Assessment and Reporting** Structured penetration-testing workflow; interpreting scanner results, false-positive checking, risk categorization, evidence, remediation recommendations and retesting. Reference: R3: Ch. 20–21; R4: Ch. 20–21	**3**	4
. 5	5.1	**Practical Exercises and Labs** 1. **Isolated Lab Setup & Kali Linux Fundamentals:** Configure an isolated Kali Linux VM and a local vulnerable web target (DVWA / OWASP Juice Shop) on a host only network; document scope and	30	5

MGU-BCA (HONOURS) Syllabus

		Rules of Engagement; practice shell navigation, file search, grep/sed, permissions, apt, and process inspection. 2. **Network Inspection, Apache Service & HTTP Protocol Analysis:** Inspect local network interfaces and DNS settings; manage a local Apache web service; capture and analyze HTTP requests, responses, headers, cookies, and sessions using Browser DevTools and cURL. 3. **Authorized Host Discovery, Port Scanning & Fingerprinting**: Perform host discovery, port scanning, and service/version identification against the lab target using Nmap; fingerprint web technologies and check server misconfigurations using WhatWeb and Nikto. 4. **Interception Proxy Configuration & Attack-Surface Mapping**: Configure Burp Suite Community Edition or OWASP ZAP; intercept, inspect, and replay HTTP/API requests; enumerate application pages, hidden parameters, cookies, and REST endpoints. 5. **SQL Injection & Command Injection Verification and Fix:** Verify SQL injection and command injection in the lab sandbox using safe manual test payloads (and supervised SQLMap); implement and verify remediation using parameterized queries / prepared statements. 6. **Cross-Site Scripting (XSS) Identification & CSP Mitigation:** Identify reflected, stored, and DOM-based XSS flaws in a training application; demonstrate remediation through context-aware		

MGU-BCA (HONOURS) Syllabus

		output encoding and a restrictive Content Security Policy (CSP) header. 7. **CSRF & XML/XXE Vulnerability Assessment:** Reproduce a Cross-Site Request Forgery (CSRF) state change and evaluate a controlled XXE / unsafe-XML parser scenario; verify anti-CSRF token, SameSite cookie, and secure XML parser mitigations. 8. **Authentication, Session & Broken Access Control Testing:** Test multi-role lab accounts for login weaknesses, insecure cookie flags, session fixation, and horizontal/vertical access-control flaws (IDOR/BOLA); recommend and verify configuration/code fixes. 9. **Automated Vulnerability Scanning & False-Positive Triage:** Execute an automated OWASP ZAP assessment against an approved local target; manually validate reported alerts, filter false positives, and assign CVSS severity ratings. 10. **End-to-End Penetration Test :** Conduct a supervised mini assessment on an assigned local web application: capture evidence logs/screenshots, rank risk and retest applied fixes. **Report: Students should prepare a** formal executive and technical report on End-to-end Penetration Test and submit for end semester practical evaluation.		

MGU-BCA (HONOURS) Syllabus

Teaching and Learning Approach	**Classroom Procedure (Mode of transaction)** • Lecture • Lab Sessions • Collaborative learning • Demonstration
Assessment Types	**MODE OF ASSESSMENT** **A. Continuous Comprehensive Assessment (CCA)** **CCA for Theory: 30 Marks** ● Written test ● Assignments ● Seminar ● MCQ/Viva **CCA for Practical: 15 Marks** ● Practical assignments ● Observation of practical skills ● Viva etc.
	B. End Semester Evaluation (ESE) **ESE for Theory: Written Test (70 Marks, 2 Hrs)** Part A: Very Short Answer Questions. (Answer all) – (5 x 2 = 10 marks) Part B: Short answer questions (5 out of 7 Questions) – (5 x 6 = 30 marks) Part C: Essay questions (2 out of 3 Questions)- (2 x 15 = 30 marks) **ESE for Practical: (35 Marks)** 1. Report - 15 Marks 2. Viva - 20 Marks

MAHATMA GANDHI UNIVERSITY · KOTTAYAM ·
MGU-BCA (HONOURS)
Syllabus

References:

1. Andrew Hoffman, Web Application Security: Exploitation and Countermeasures for Modern Web Applications, 2nd Edition, O'Reilly Media, 2024. (Module 1 - 4)
2. Dafydd Stuttard and Marcus Pinto, The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws, 2nd Edition, Wiley, 2011 (Module 1-3) (Module 1-4)
3. Malcolm McDonald, Web Security for Developers: Real Threats, Practical Defense, No Starch Press, 2020. (Module 1, Module3, Module 4)
4. OccupyTheWeb, Linux Basics for Hackers: Getting Started with Networking, Scripting, and Security in Kali, 2nd Edition, No Starch Press, 2025. (Module 1-2)

Resources for Current Standards & Labs:

1. OWASP Top 10:2025 (Current web risk taxonomy) & OWASP Application Security Verification Standard (ASVS) (Defensive verification controls).
2. OWASP Web Security Testing Guide (WSTG) (Structured manual testing methodology) & Kali Linux Official Documentation (Tool manuals).
3. PortSwigger Web Security Academy & OWASP Juice Shop / WebGoat / DVWA (Free interactive and local sandbox)

MGU-BCA (HONOURS)

Syllabus