COURSE TITLE	**BACKEND APPLICATION DEVELOPMENT**
COURSE CODE	**05FN0404**
COURSE CREDITS	**4**

Objective:

1 To impart a thorough understanding about Node Basics.
2 To impart a thorough understanding about Express.js in detail.
3 To impart a thorough understanding about MVC in detail.
4 To impart through understanding about database in SQL and NoSQL.
5 To develop secure and scalable backend applications with authentication and API integration.

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

1 Students will gain a comprehensive understanding of Node.js fundamentals including creating servers, handling requests, and managing asynchronous operations using event-driven architecture.
2 Students will become proficient in using Express.js framework for building web applications, including implementing routing, middleware, and error handling.
3 Students will understand different database options for Node.js applications, including SQL and NoSQL databases, and gain hands-on experience in integrating databases with Node.js using Sequelize ORM and other libraries.
4 Students will learn to integrate frontend technologies like HTML, CSS, and JavaScript with Node.js and Express.js for building full-stack web applications.

Pre-requisite of course:NA

Teaching and Examination Scheme

Theory Hours	Tutorial Hours	Practical Hours	ESE	IA	CSE	Viva	Term Work
3	0	2	0	30	20	50	50

DR. SUNIL LALCHAND BAJEJA
Digitally signed by (Name of HOD)

DR. R SRIDARAN
Digitally signed by (Name of Dean/ Principal)

Contents : Unit	Topics	Contact Hours
1	**Node Basics** Creating node server, Node life cycle and event loop,understanding requests, sending response,Routing request,Redirect requests,Parsing request body, understanding event driven execution,Blocking and non-blocking code, Node.js looking behind the scenes, Understanding NPM scripts, installing 3rd party packages, using nodemon for auto restarts, understanding different types of errors, Finding and fixing syntax error, Dealing with runtime errors, Dealing with runtime errors,Using the debugger, Restarting the debuggerautomatically after editingapp,Logical errors,Using the debugger,Restarting the debugger automatically after editing	11
2	**Express** What is Express.js, Installing Express.js, Adding Middleware,Parsing Incoming requests,Limiting Middleware Execution to POST request, Using Express router,Adding a 404 page, filtering paths, creating HTML pages, Serving HTML pages, returning 404 page, using a helper function for navigation, Styling pages, serving files statically, Sharing data across Requests & users, Templating Engines overview, Installing and implementing Pug, Outputting dynamic content, Converting HTML files to Pug,Adding a layout, Finishing the Pug template, working with handlebars, Adding a layout to handle bars, Working with EJS, Working on the layout with Partials	11
3	**MVC** What is MVC, Adding controller, Adding a product Model, storing data in the files via the Model, Refactoring the file storage code, Creating the shop structure,Working on navigation,Storing product data,displaying product data,Editing and deleting products,Adding another item,Adding product ID to the path, extracting dynamic params,Loading product details data,Rendering the product detail view,Passing data with POST requests, Adding cart model, Using query parameters, Pre populating the edited product page with Data,linking to the edit page,Editing the product data, Adding the product delete functionality,Deleting cart items,Displaying cart items on the cart page, fixing delete product bugs	11
4	**My SQL & NoSQL** Choosing database, NoSQL intro, comparing NoSQL and SQL, setting up MySQL, Connecting app to SQL database, Basic SQL and creating a table, retrieving data, Fetching products, Inserting data into database, fetching a single product with the where condition,What is Sequalize, connecting to a database, defining a model, syncing JS definition to a database, Inserting Data & creating a product, retrieving data & finding products, getting a single product with the where clause, fetching admin products,Updating products, deleting products, creating a user manual, Adding a one to many relationship	12
Total Hours		**45**

DR. SUNIL LALCHAND BAJEJA
Digitally signed by (Name of HOD)

DR. R SRIDARAN
Digitally signed by (Name of Dean/ Principal)

Suggested List of Experiments:

Contents : Unit	Topics	Contact Hours
1	**Unit-1** Write a Node.js server that listens on a specified port and responds with "Hello World" for every request,Implement routing in Node.js to handle different types of requests such as GET, POST, PUT, DELETE,Create a Node.js server that redirects requests from one URL to another, Parse request bodies in Node.js and log the parsed data to the console,Write a script to demonstrate the difference between blocking and non-blocking code execution in Node.js, Debug a Node.js application using the built-in debugger,Write error-handling middleware in Express.js to catch and handle runtime errors,Implement automatic restarting of the debugger after editing a Node.js application, Implement a middleware in Express.js to authenticate user requests using JWT (JSON Web Tokens),Develop a file upload feature in a Node.js application using Multer middleware,Explore the event loop in Node.js and describe its significance in asynchronous programming	30
2	**Unit-2** Create and publish an NPM package with a simple functionality,Write unit tests for a Node.js application using a testing framework like Mocha and assertion library like Chai,Explore and implement server-side rendering (SSR) in Node.js using frameworks like Next.js or Nuxt.js, Create a RESTful API endpoint in Node.js to perform CRUD operations on a resource like 'todos' or 'users',Set up and configure a reverse proxy server using Nginx to forward requests to a Node.js application, Implement rate limiting for API endpoints in a Node.js application to prevent abuse or DoS attacks,Integrate a logging framework like Winston or Bunyan into a Node.js application to track and analyze server-side events., Develop a web scraping script in Node.js to extract data from a website and save it to a local database or file,Create a command-line interface (CLI) tool in Node.js using libraries like Commander.js for parsing arguments and executing commands	30
Total Hours		**60**

Textbook :

1 Node.js Design Patterns, Mario Casciaro, Packt Publishing, 2014

2 Pro Express.js: Mastering the Fundamentals of Web Development with Node.js, Azat Mardan, Apress, 2014

References:

1 Express in Action, Express in Action, Evan Hahn, Manning Publications, 2016

2 Web Development with Node and Express: Leveraging the JavaScript Stack, Web Development with Node and Express: Leveraging the JavaScript Stack, Ethan Brown , O'Reilly Media, 2019

Suggested Theory Distribution:

The suggested theory distribution as per Bloom's taxonomy is as follows. This distribution serves as guidelines for teachers and students to achieve effective teaching-learning process

Distribution of Theory for course delivery					
Remember / Knowledge	**Understand**	**Apply**	**Analyze**	**Evaluate**	**Higher order Thinking / Creative**
20.00	30.00	25.00	15.00	10.00	0.00

Instructional Method:

1 PPT,BOARD WORK,PRACTICALS

Supplementary Resources:

1 https://nodejs.org/en/docs/
2 https://expressjs.com/
3 https://learn.mongodb.com
4 https://developer.mozilla.org/en-US/docs/Web/JavaScript
5 https://www.theodinproject.com/
6 https://www.youtube.com/@TraversyMedia

DR. SUNIL LALCHAND BAJEJA
Digitally signed by (Name of HOD)

DR. R SRIDARAN
Digitally signed by (Name of Dean/ Principal)