ABES

Estd. 2000

Full Stack Development: FSD

Semester –Even/Odd, Session 2024-25

Day-0

Full Stack Developer

Over 2 billion Websites

Approximately 4.54 billion active internet users

Need for enhanced customer experience

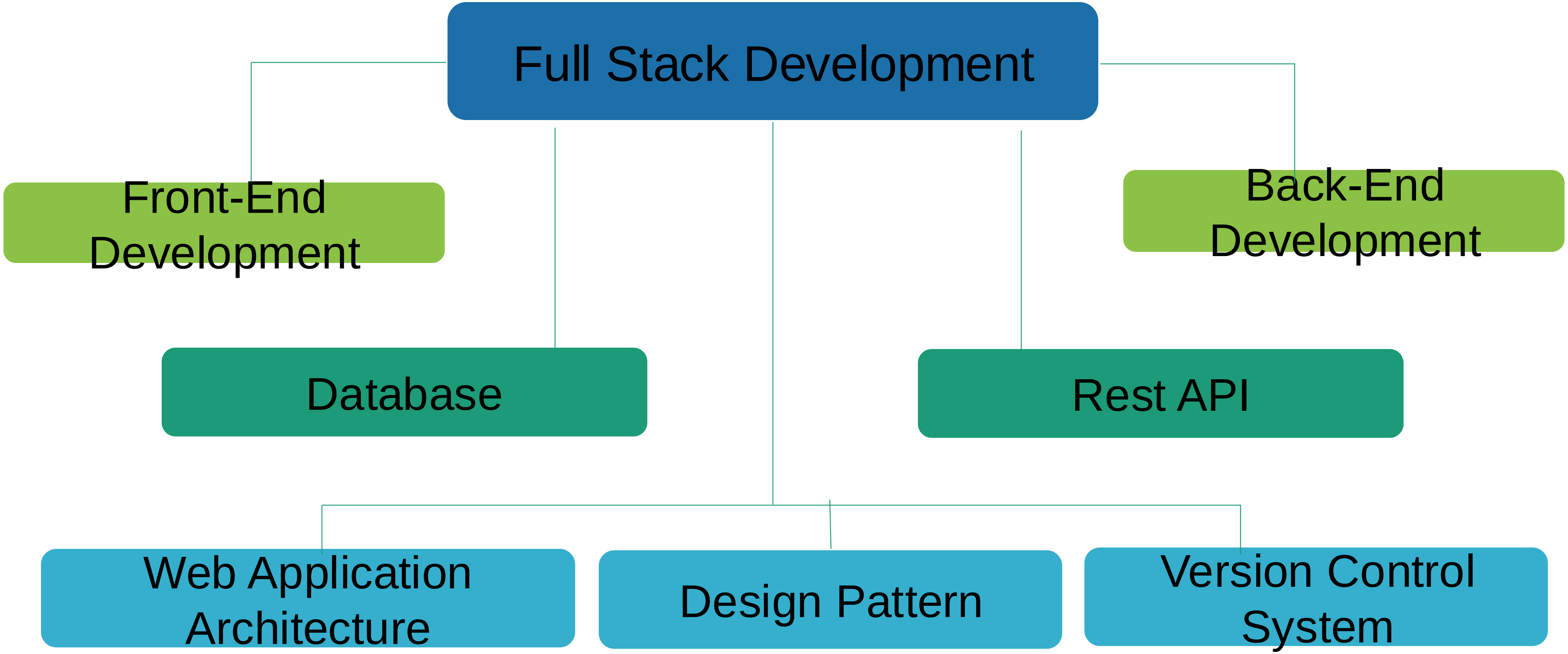
The Full Stack developer role is the No.1 position to be filled in 2020 according to 38% of hiring managers. This is closely followed by the role of a back-end developer.

- Among one of the most in demand job profiles for B. Tech Graduates
- Ability to work on both frond-end and back-end
- Knowledge of a broad spectrum of technologies

- Easy to learn
- Provides a much wider understanding of the scope of the project
- Creative flexibility
- Handsome salary packages

The demand for full-stack developers will steadily increase in the market as new technologies emerge.

Components of Full Stack Development



Full Stack Developer – Job Roles/Job Profiles

ABES
Estd. 2000

1. **Front-End Developer –** Front-end web development, also known as client-side development is the practice of producing HTML, CSS and JavaScript for a website or Web Application.

2. **Back-End Developer –** Back-end Development refers to the server-side development. It focuses on databases, scripting, website architecture.

3. **Full Stack Developer –** Full Stack Development requires proficiency in both front-end and back-end development.

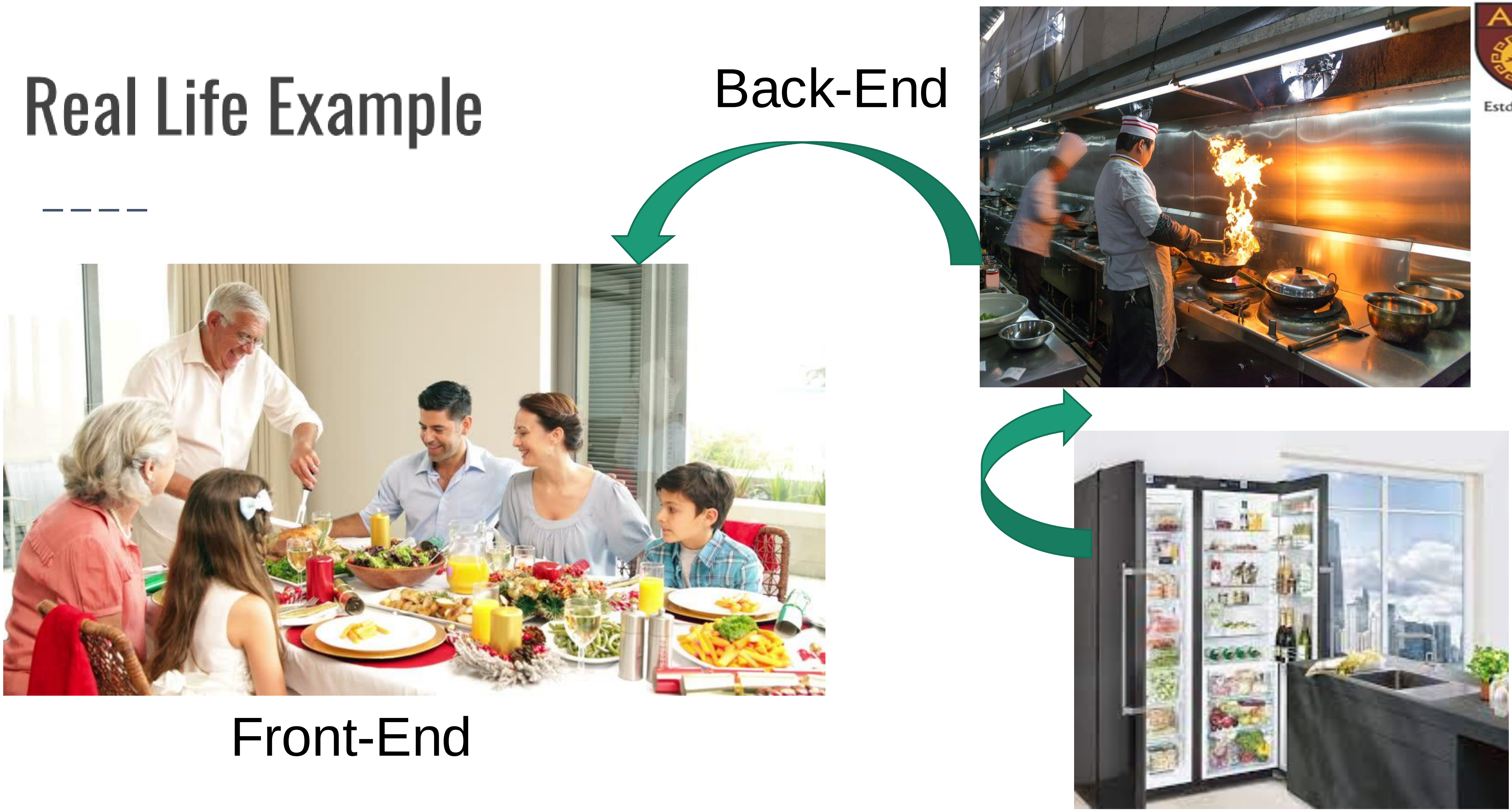
4. **User Experience designer –** Designing interactions between system and user.

5. **Software Developer/Engineer –** A generalized profile usually it may refer to any of the above job requirement.

6. **Web Designer –** Web designer is the role of someone who formulates solutions to problems specific to the web environment based on the higher-level limitations of that environment.

7. **DevOps engineer –** Deploying the system to production servers, maintaining the system, creating and maintaining application development and production environments, and preparing for backup and recovery.

Real Life Example

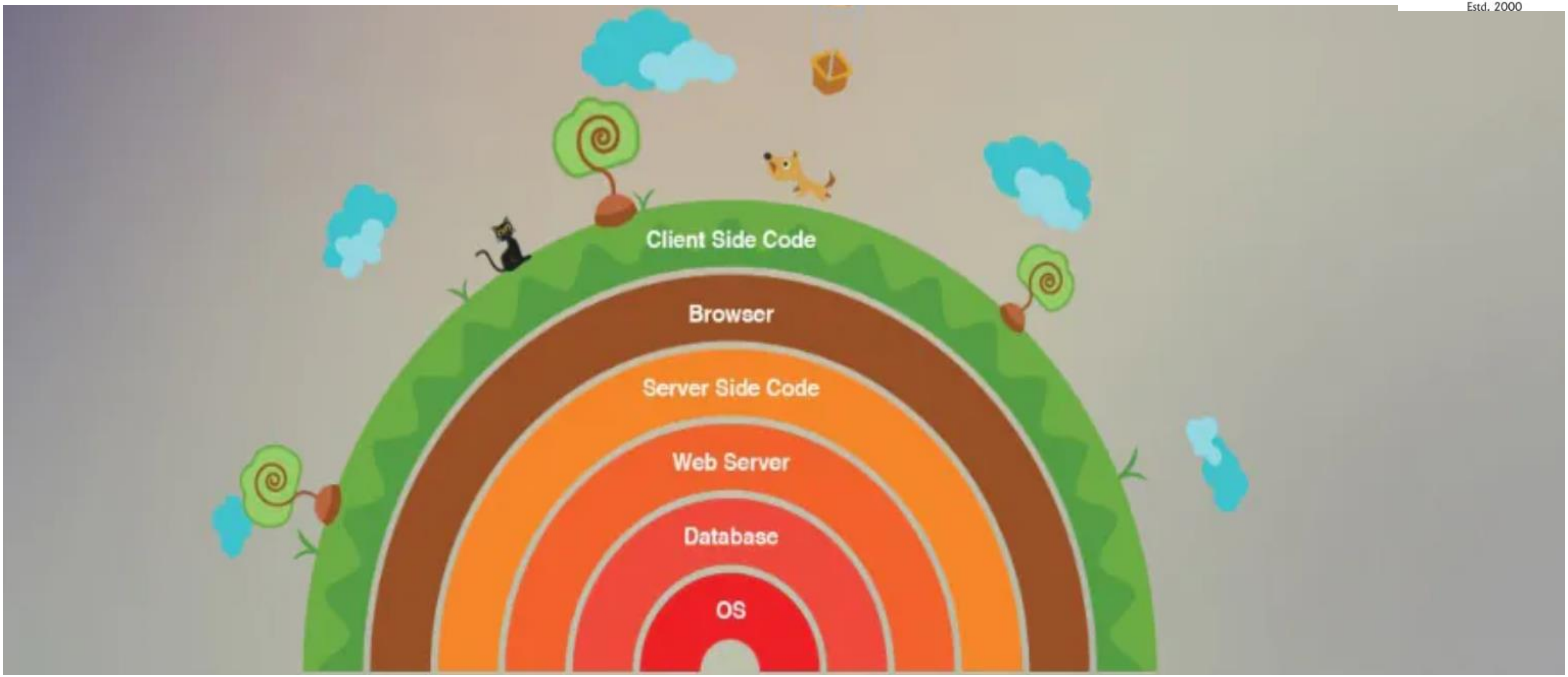


Back-End

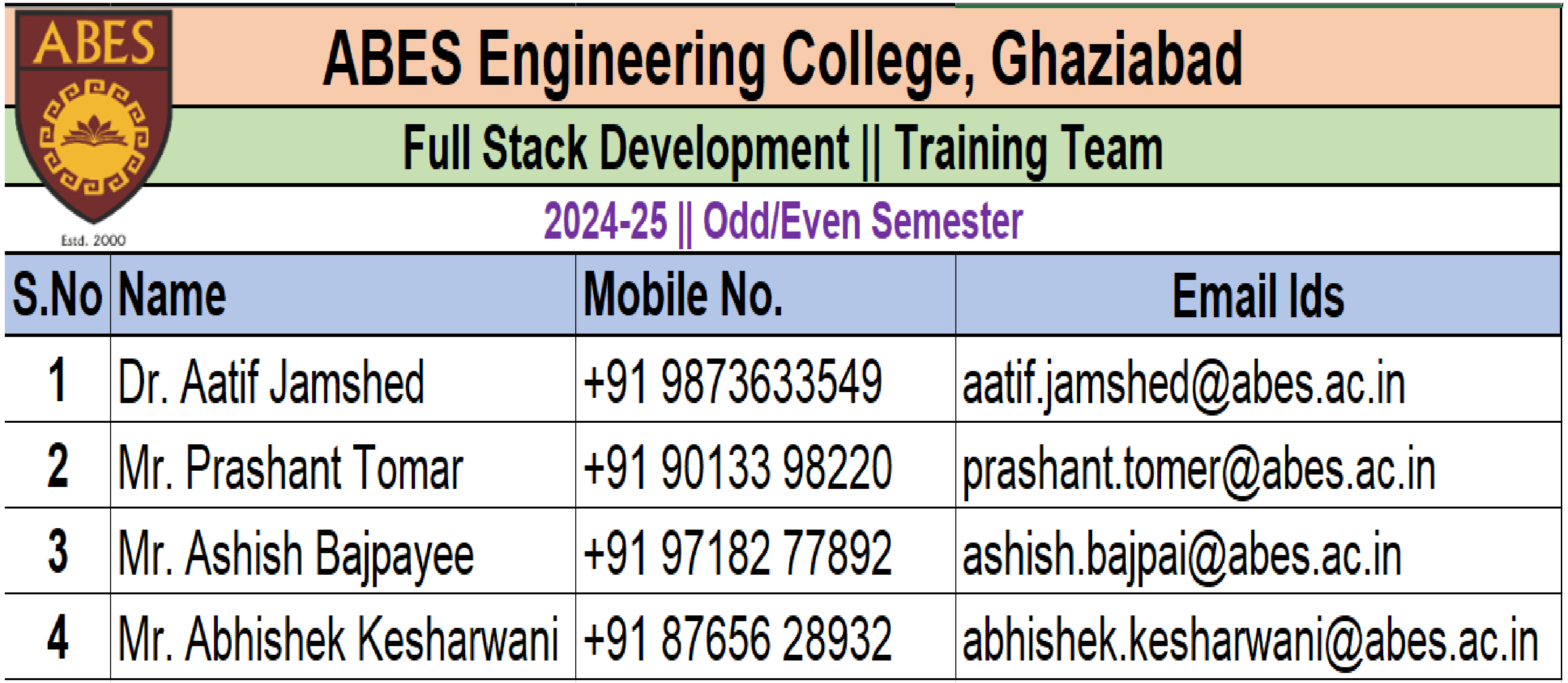
Front-End

Database

Layers in Full Stack Development



Training Team



ABES Engineering College, Ghaziabad

Full Stack Development || Training Team

2024-25 || Odd/Even Semester

S.No	Name	Mobile No.	Email Ids
1	Dr. Aatif Jamshed	+91 9873633549	aatif.jamshed@abes.ac.in
2	Mr. Prashant Tomar	+91 90133 98220	prashant.tomer@abes.ac.in
3	Mr. Ashish Bajpayee	+91 97182 77892	ashish.bajpai@abes.ac.in
4	Mr. Abhishek Kesharwani	+91 87656 28932	abhishek.kesharwani@abes.ac.in

Objectives

- To equip students with the skills to create responsive, user-friendly web interfaces using HTML, CSS, and JavaScript frameworks like React.js.

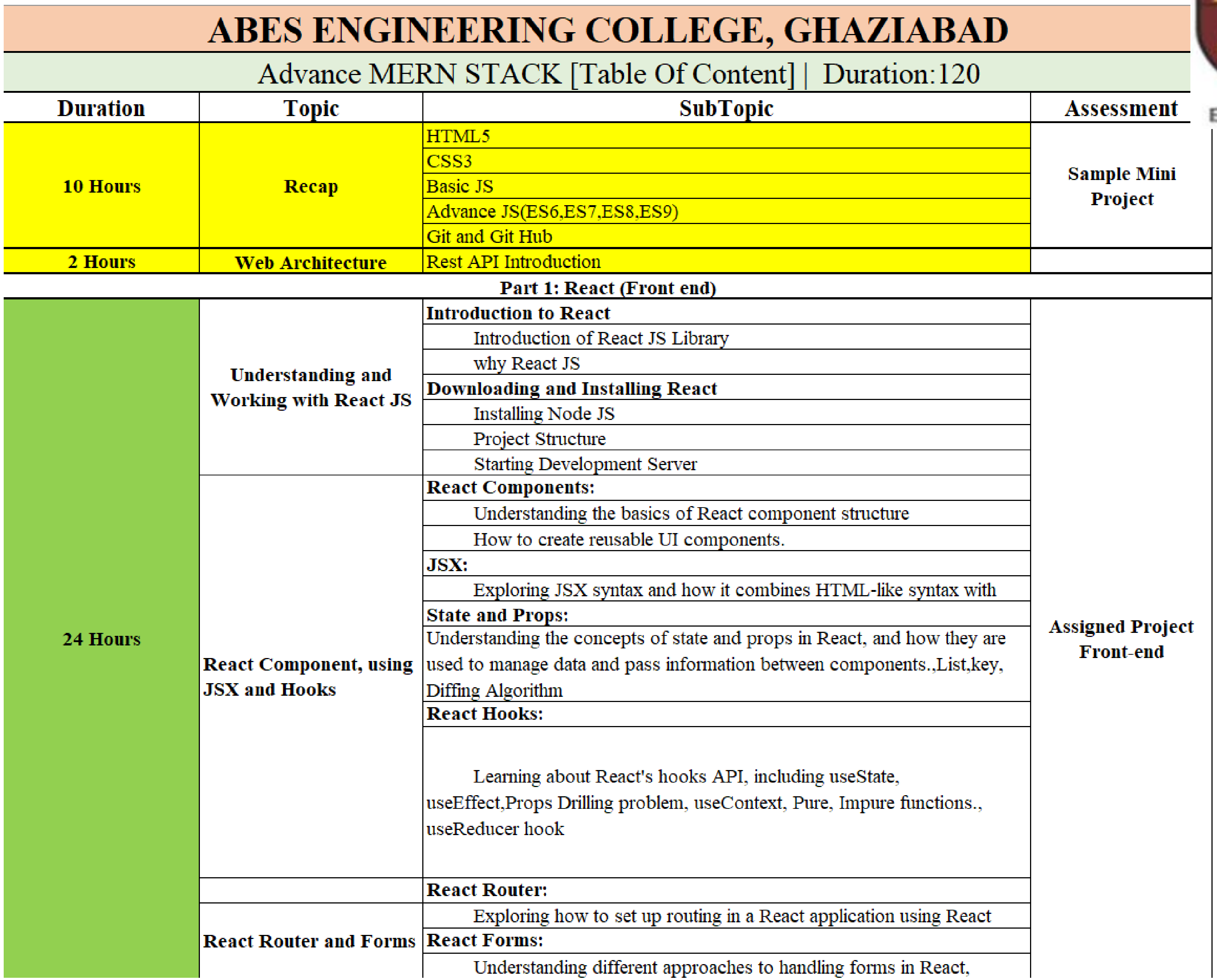
- To enable students to build robust server-side applications using Node.js and Express.js. They will learn to manage server operations, handle requests and responses, and integrate with databases.

- To gain the ability to perform CRUD operations, model data, and connect the database with the application back-end.

- To implement and manage API endpoints, handle different HTTP methods, and ensure secure data transmission.

- To learn deployment strategies using cloud services and DevOps tools, ensuring their applications are production-ready.

Syllabus

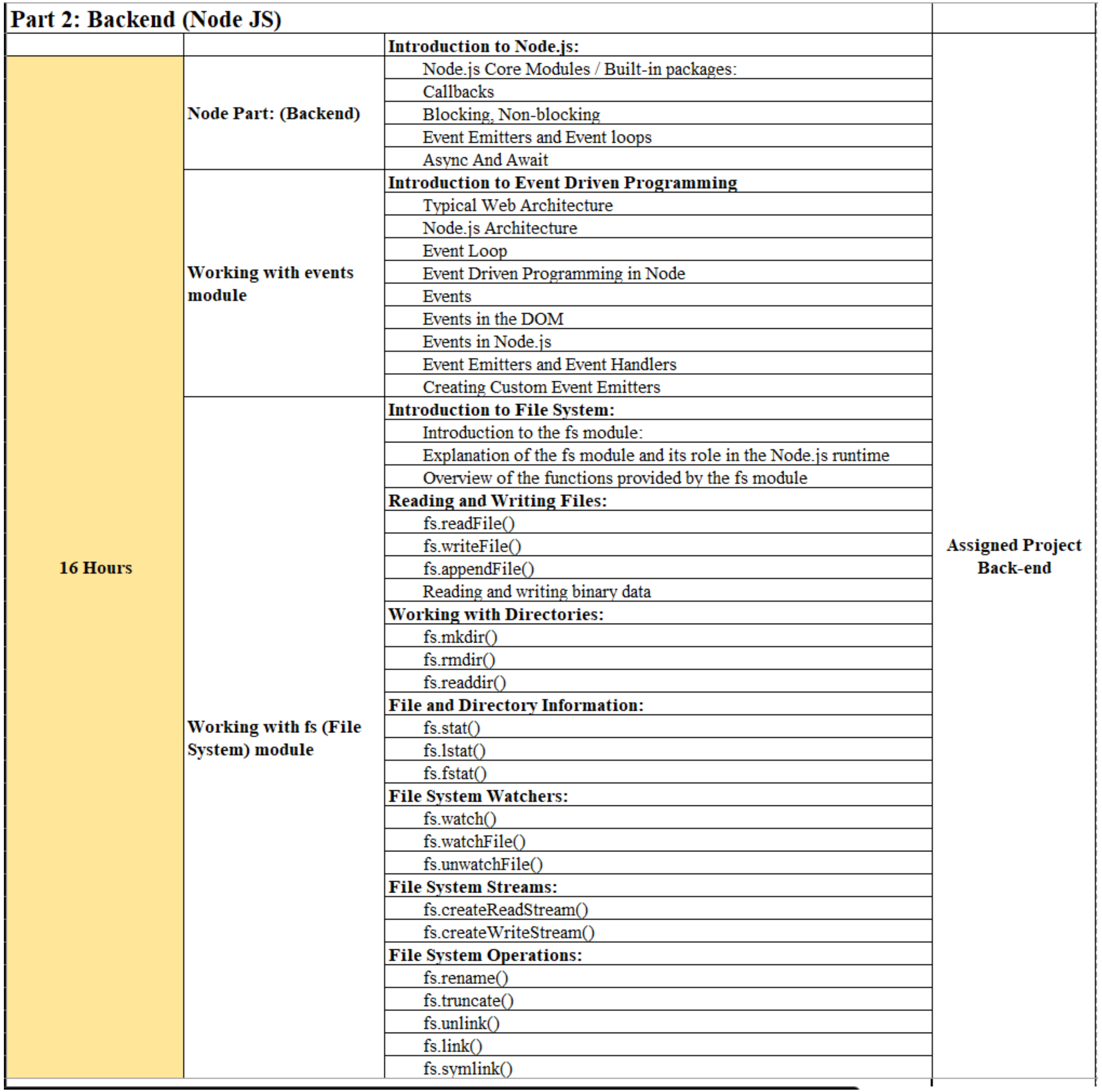


ABES ENGINEERING COLLEGE, GHAZIABAD

Advance MERN STACK [Table Of Content] | Duration:120

Duration	Topic	SubTopic	Assessment
10 Hours	**Recap**	HTML5	**Sample Mini Project**
		CSS3	
		Basic JS	
		Advance JS(ES6,ES7,ES8,ES9)	
		Git and Git Hub	
2 Hours	**Web Architecture**	Rest API Introduction	
		Part 1: React (Front end)	
24 Hours	**Understanding and Working with React JS**	**Introduction to React**	**Assigned Project Front-end**
		Introduction of React JS Library	
		why React JS	
		Downloading and Installing React	
		Installing Node JS	
		Project Structure	
		Starting Development Server	
	React Component, using JSX and Hooks	**React Components:**	
		Understanding the basics of React component structure	
		How to create reusable UI components.	
		JSX:	
		Exploring JSX syntax and how it combines HTML-like syntax with	
		State and Props:	
		Understanding the concepts of state and props in React, and how they are used to manage data and pass information between components.,List,key, Diffing Algorithm	
		React Hooks:	
		Learning about React's hooks API, including useState, useEffect,Props Drilling problem, useContext, Pure, Impure functions., useReducer hook	
		React Router:	
	React Router and Forms	Exploring how to set up routing in a React application using React	
		React Forms:	
		Understanding different approaches to handling forms in React,	

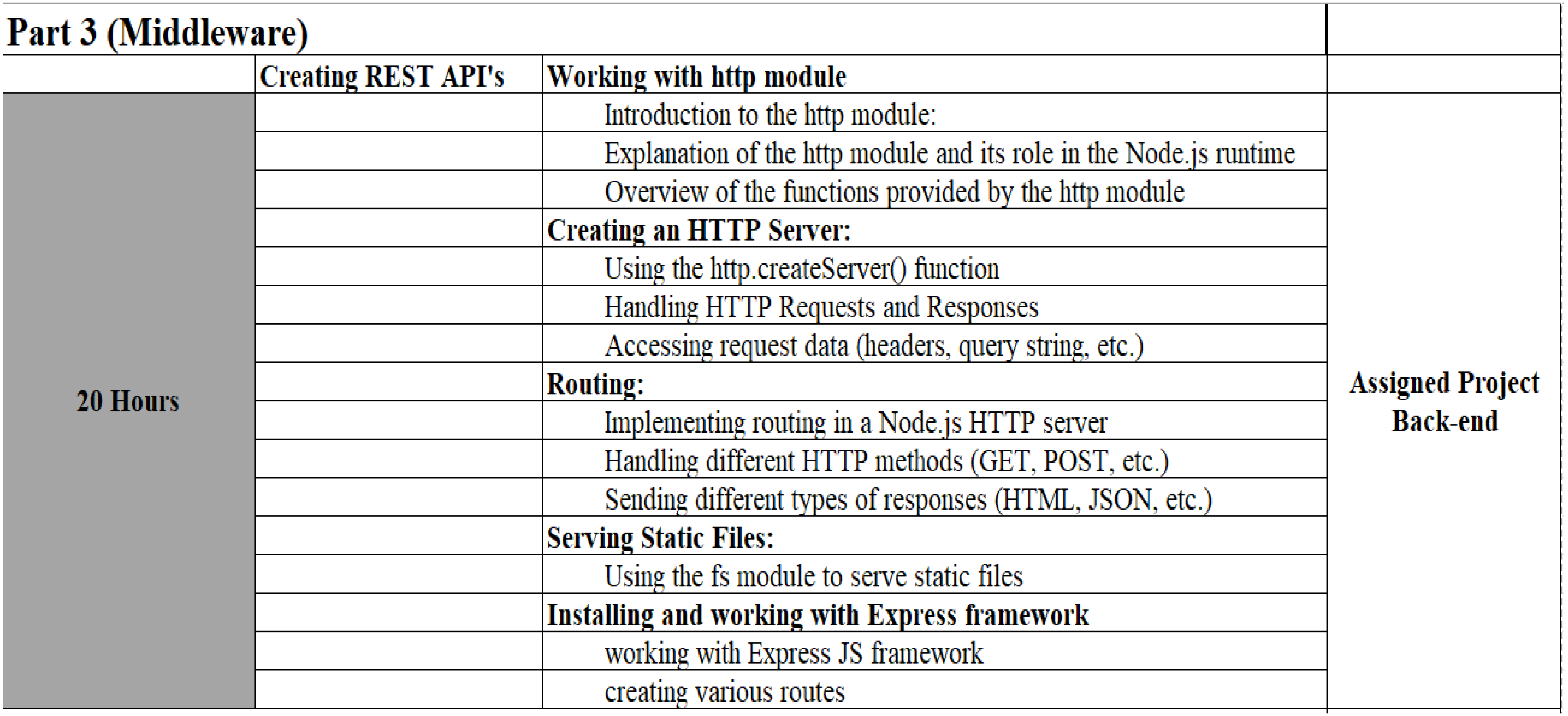
Syllabus



Part 2: Backend (Node JS)			
		Introduction to Node.js:	**Assigned Project Back-end**
16 Hours	**Node Part: (Backend)**	Node.js Core Modules / Built-in packages:	
		Callbacks	
		Blocking, Non-blocking	
		Event Emitters and Event loops	
		Async And Await	
	Working with events module	**Introduction to Event Driven Programming**	
		Typical Web Architecture	
		Node.js Architecture	
		Event Loop	
		Event Driven Programming in Node	
		Events	
		Events in the DOM	
		Events in Node.js	
		Event Emitters and Event Handlers	
		Creating Custom Event Emitters	
	Working with fs (File System) module	**Introduction to File System:**	
		Introduction to the fs module:	
		Explanation of the fs module and its role in the Node.js runtime	
		Overview of the functions provided by the fs module	
		Reading and Writing Files:	
		fs.readFile()	
		fs.writeFile()	
		fs.appendFile()	
		Reading and writing binary data	
		Working with Directories:	
		fs.mkdir()	
		fs.rmdir()	
		fs.readdir()	
		File and Directory Information:	
		fs.stat()	
		fs.lstat()	
		fs.fstat()	
		File System Watchers:	
		fs.watch()	
		fs.watchFile()	
		fs.unwatchFile()	
		File System Streams:	
		fs.createReadStream()	
		fs.createWriteStream()	
		File System Operations:	
		fs.rename()	
		fs.truncate()	
		fs.unlink()	
		fs.link()	
		fs.symlink()	

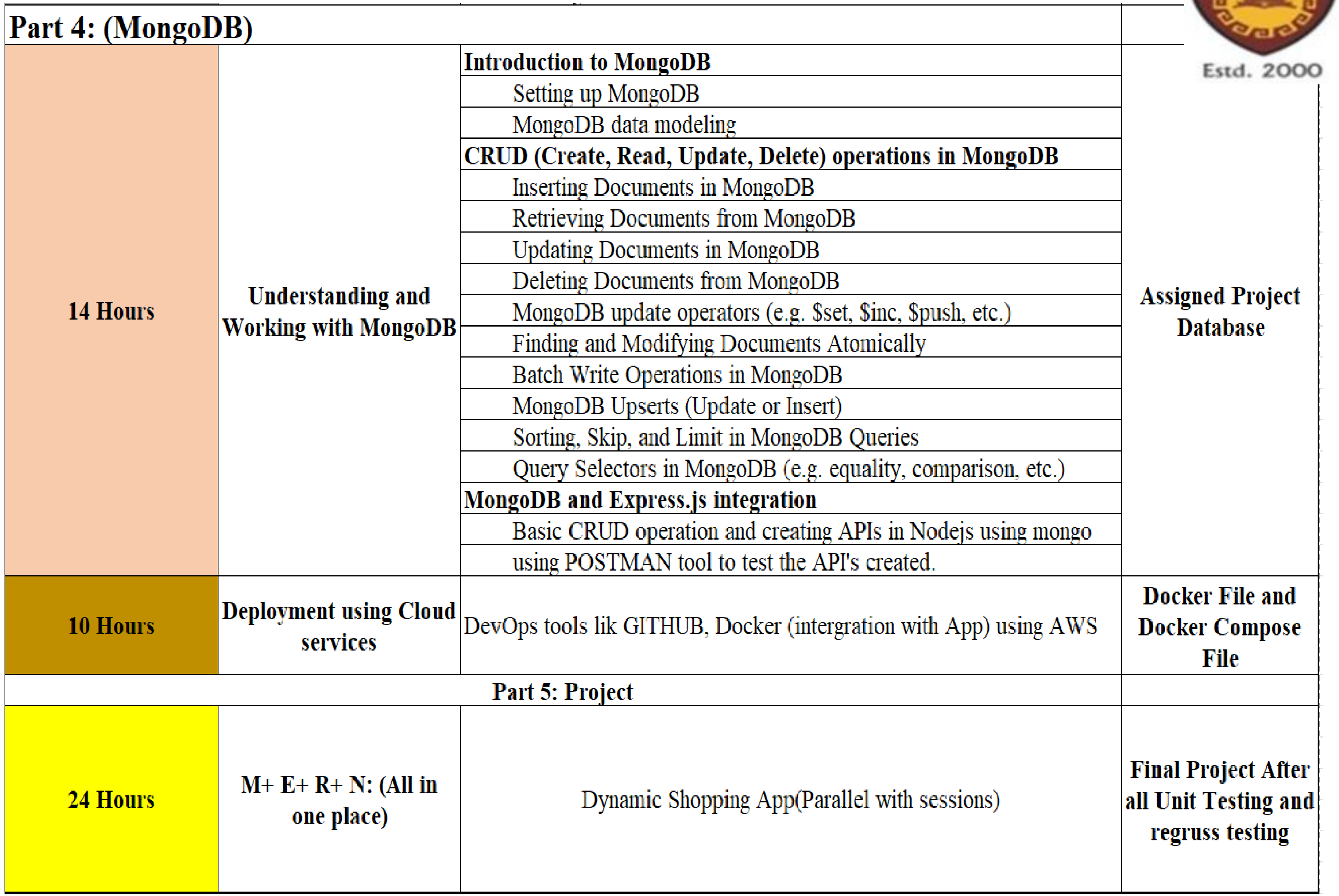
Estd. 2000

Syllabus



Part 3 (Middleware)			
	Creating REST API's	**Working with http module**	
20 Hours		Introduction to the http module:	**Assigned Project Back-end**
		Explanation of the http module and its role in the Node.js runtime	
		Overview of the functions provided by the http module	
		Creating an HTTP Server:	
		Using the http.createServer() function	
		Handling HTTP Requests and Responses	
		Accessing request data (headers, query string, etc.)	
		Routing:	
		Implementing routing in a Node.js HTTP server	
		Handling different HTTP methods (GET, POST, etc.)	
		Sending different types of responses (HTML, JSON, etc.)	
		Serving Static Files:	
		Using the fs module to serve static files	
		Installing and working with Express framework	
		working with Express JS framework	
		creating various routes	

Syllabus



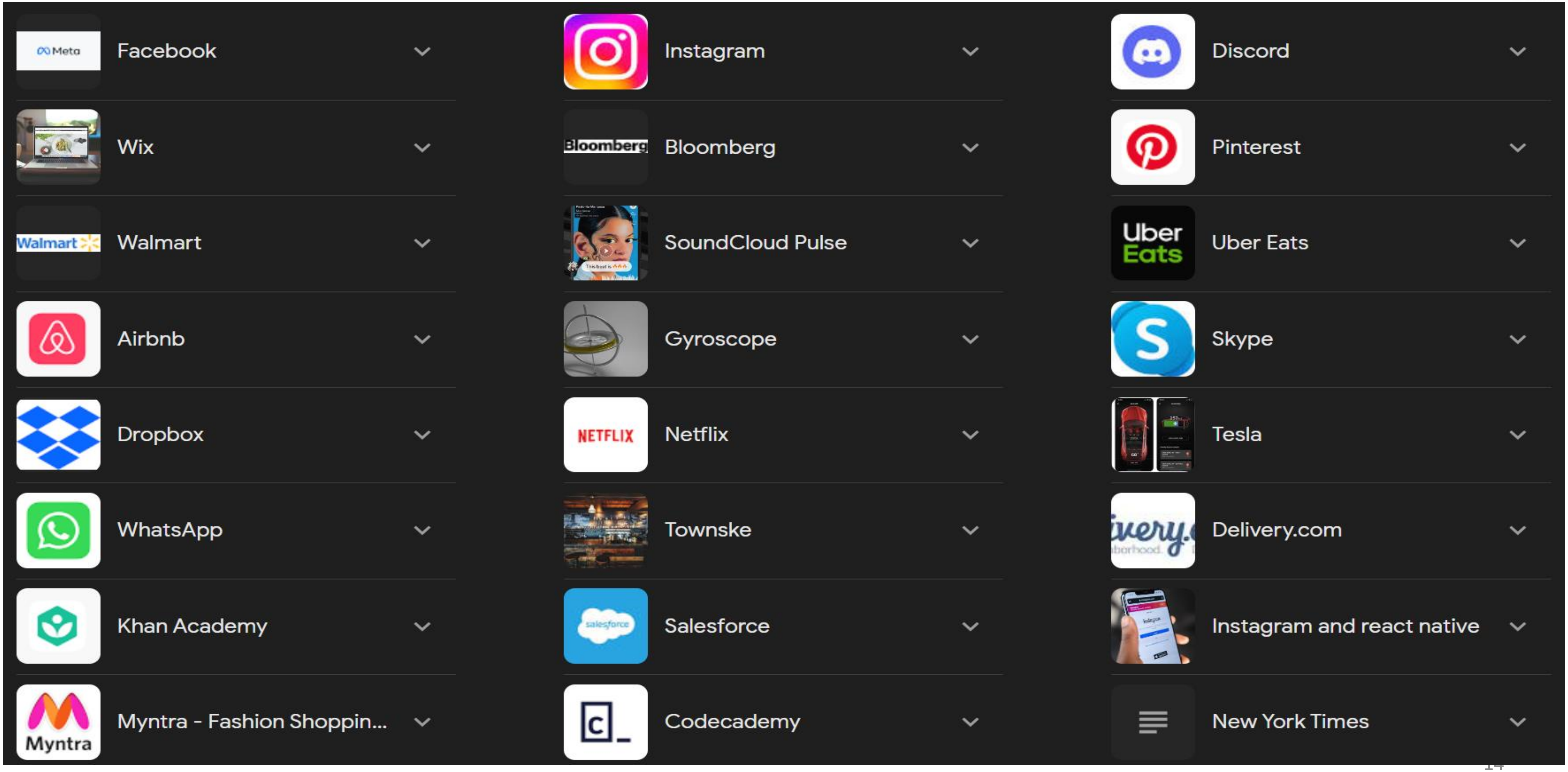
Part 4: (MongoDB)			
14 Hours	**Understanding and Working with MongoDB**	**Introduction to MongoDB**	**Assigned Project Database**
		Setting up MongoDB	
		MongoDB data modeling	
		CRUD (Create, Read, Update, Delete) operations in MongoDB	
		Inserting Documents in MongoDB	
		Retrieving Documents from MongoDB	
		Updating Documents in MongoDB	
		Deleting Documents from MongoDB	
		MongoDB update operators (e.g. $set, $inc, $push, etc.)	
		Finding and Modifying Documents Atomically	
		Batch Write Operations in MongoDB	
		MongoDB Upserts (Update or Insert)	
		Sorting, Skip, and Limit in MongoDB Queries	
		Query Selectors in MongoDB (e.g. equality, comparison, etc.)	
		MongoDB and Express.js integration	
		Basic CRUD operation and creating APIs in Nodejs using mongo	
		using POSTMAN tool to test the API's created.	
10 Hours	**Deployment using Cloud services**	DevOps tools lik GITHUB, Docker (intergration with App) using AWS	**Docker File and Docker Compose File**
		Part 5: Project	
24 Hours	**M+ E+ R+ N: (All in one place)**	Dynamic Shopping App(Parallel with sessions)	**Final Project After all Unit Testing and regruss testing**

Outcomes

Upon completion of the cour students will be able to:

- Understand the basics of full stack web development
- Understand the basic of Web Application Architecture
- Develop responsive web pages using HTML and CSS
- Implement client-side scripting using JavaScript
- Build single page application using React JS
- Develop RestAPI using Node JS and ExpressJS.
- Manage data using MongoDB
- Deploy web applications on NGINIX Server
- Understand the best practices for building secure web applications
- Develop real-world web applications using various technologies learned in the course
- Demonstrate your technical proficiency using Git Repository

Popular apps built with React



- Facebook
- Wix
- Walmart
- Airbnb
- Dropbox
- WhatsApp
- Khan Academy
- Myntra - Fashion Shoppin...
- Instagram
- Bloomberg
- SoundCloud Pulse
- Gyroscope
- Netflix
- Townske
- Salesforce
- Codecademy
- Discord
- Pinterest
- Uber Eats
- Skype
- Tesla
- Delivery.com
- Instagram and react native
- New York Times

Thank you