**Department of Computer Science & Enginnering**

**PYTHON: DEEP LEARNING (Skill Oriented Course) (R20)**

**LIST OF EXPERIMENTS**

**Size- (2\*3) Count- 1**

|  |  |
| --- | --- |
| Exercise-1 | **Module name :** Build a Convolution Neural Network for Image Recognition. |
| Go through the modules of the course mentioned and answer the self-assessment questions given in the link below at the end of the course. |
| Exercise-2 | **Module name :** Understanding and Using ANN : Identifying age group of an actor. |
| **Exercise :** Design Artificial Neural Networks for Identifying and Classifying an actor using Kaggle Dataset. |
| Exercise-3 | **Module name :** Understanding and Using CNN : Image recognition . |
| **Exercise**: Design a CNN for Image Recognition which includes hyperparameter tuning. |
| Exercise-4 | **Module name :** Predicting Sequential Data. |
| **Exercise:** Implement a Recurrence Neural Network for Predicting Sequential Data. |
| Exercise-5 | **Module Name:** Removing noise from the images . |
| **Exercise:** Implement Multi-Layer Perceptron algorithm for Image denoising hyperparameter tuning. |
| Exercise-6 | **Module Name:** Advanced Deep Learning Architectures . |
| **Exercise:** Implement Object Detection Using YOLO. |
| Exercise-7 | **Module Name:** Optimization of Training in Deep Learning. |
| **Exercise Name:** Design a Deep learning Network for Robust Bi-Tempered Logistic Loss. |
| Exercise-8 | **Module name:** Advanced CNN . |
| **Exercise:** Build AlexNet using Advanced CNN. |
| Exercise-9 | **Module name:** Autoencoders Advanced. |
| **Exercise:** Demonstration of Application of Autoencoders. |
| Exercise-10 | **Module name:** Advanced GANs . |
| **Exercise:**Demonstration of GAN. |
| Exercise-11 | **Module name :** Capstone project |
| **Exercise :** Complete the requirements given in capstone project. |
| Exercise-12 | **Module name :** Capstone project . |
| **Exercise :** Complete the requirements given in capstone project. |

**Department of Computer Science & Enginnering**

**COMPILER DESIGN LAB (R20)**

**LIST OF EXPERIMENTS**

**Size- (2\*3) Count- 1**

|  |  |
| --- | --- |
| Exercise-1 | Write a C program to identify different types of Tokens in a given Program. |
| Exercise-2 | Write a Lex Program to implement a Lexical Analyzer using Lex tool. |
| Exercise-3 | Write a C program to Simulate Lexical Analyzer to validating a given input String. |
| Exercise-4 | Write a C program to implement the Brute force technique of Top down Parsing. |
| Exercise-5 | Write a C program to implement a Recursive Descent Parser. |
| Exercise-6 | Write C program to compute the First and Follow Sets for the given Grammar. |
| Exercise-7 | Write a C program for eliminating the left recursion and left factoring of a given grammar. |
| Exercise-8 | Write a C program to check the validity of input string using Predictive Parser. |
| Exercise-9 | Write a C program for implementation of LR parsing algorithm to accept a given input string. |
| Exercise-10 | Write a C program for implementation of a Shift Reduce Parser using Stack Data Structure to accept a given input string of a given grammar. |
| Exercise-11 | Simulate the calculator using LEX and YACC tool. |
| Exercise-12 | Generate YACC specification for a few syntactic categories. |
| Exercise-13 | Write a C program for generating the three address code of a given expression/statement. |
| Exercise-14 | Write a C program for implementation of a Code Generation Algorithm of a given expression/statement. |

**Department of Computer Science & Enginnering**

**CRYPTOGRAPHY NETWORK SECURITY LAB (R20)**

**LIST OF EXPERIMENTS**

**Size- (2\*3) Count- 1**

|  |  |
| --- | --- |
| Exercise-1 | Write a C program that contains a string (char pointer) with a value \Hello World’. The program should XOR each character in this string with 0 and displays the result. |
| Exercise-2 | Write a C program that contains a string (char pointer) with a value \Hello World’. The program should AND or and XOR each character in this string with 127 and display the result |
| Exercise-3 | Write a Java program to perform encryption and decryption using the following algorithms:   1. Ceaser Cipher 2. Substitution Cipher 3. Hill Cipher |
| Exercise-4 | Write a Java program to implement the DES algorithm logic. |
| Exercise-5 | Write a C/JAVA program to implement the BlowFish algorithm logic. |
| Exercise-6 | Write a C/JAVA program to implement the Rijndael algorithm logic. |
| Exercise-7 | Using Java Cryptography, encrypt the text “Hello world” using BlowFish. Create your own key using Java key tool. |
| Exercise-8 | Write a Java program to implement RSA Algorithm. |
| Exercise-9 | Implement the Diffie-Hellman Key Exchange mechanism using HTML and JavaScript. Consider the end user as one of the parties (Alice) and the JavaScript application as other party (bob). |
| Exercise-10 | Calculate the message digest of a text using the SHA-1 algorithm in JAVA. |

**Department of Computer Science & Enginnering**

**MEAN STACK TECHNOLOGIES-MODULE I (HTML 5, JAVASCRIPT, EXPRESS.JS, NODE.JS AND TYPESCRIPT) (Skill Oriented Course)(R20)**

**LIST OF EXPERIMENTS**

**Size- (3\*4) Count- 1**

|  |  |  |
| --- | --- | --- |
| Exercise-1 | 1.a | **Course Name**: HTML5 - The Language |
| **Module Name:** Case-insensitivity, Platform-independency, DOCTYPE Declaration, Types of Elements, HTML Elements - Attributes, Metadata Element |
| Include the Metadata element in Homepage.html for providing description as "IEKart's is an online shopping website that sells goods in retail. This company deals with various categories like Electronics, Clothing, Accessories etc. |
| 1.b | **Course Name**: HTML5 - The Language |
| **Module Name**: Sectioning Elements |
| Enhance the Homepage.html of IEKart's Shopping Application by adding appropriate sectioning elements. |
| 1.c | **Course Name**: HTML5 - The Language |
| **Module Name**: Paragraph Element, Division and Span Elements, List Element |
| Make use of appropriate grouping elements such as list items to "About Us" page of IEKart's Shopping Application |
| 1.d | **Course Name**: HTML5 - The Language |
| **Module Name**: Link Element |
| Link "Login", "SignUp" and "Track order" to "Login.html", "SignUp.html" and "Track.html" page respectively. Bookmark each category to its details of IEKart's Shopping application |
| 1.e | **Course Name**: HTML5 - The Language |
| **Module Name**: Character Entities |
| Add the © symbol in the Home page footer of IEKart's Shopping application. |
| 1.f | **Course Name**: HTML5 - The Language |
| **Module Name**: HTML5 Global Attributes |
| Add the global attributes such as contenteditable, spellcheck, id etc. to enhance the Signup Page functionality of IEKart's Shopping application. |
| Exercise-2 | 2.a | **Course Name**: HTML5 - The Language |
| **Module Name**: Creating Table Elements, Table Elements : Colspan/Rowspan Attributes, border, cellspacing, cellpadding attributes |
| Enhance the details page of IEKart's Shopping application by adding a table element to display the available mobile/any inventories. |
| 2.b | **Course Name**: HTML5 - The Language |
| **Module Name**: Creating Form Elements, Color and Date Pickers, Select and Datalist Elements |
| Using the form elements create Signup page for IEKart's Shopping application. |
| 2.c | **Course Name**: HTML5 - The Language |
| **Module Name**: Input Elements - Attributes |
| Enhance Signup page functionality of IEKart's Shopping application by adding attributes to input elements. |
| 2.d | **Course Name**: HTML5 - The Language |
| **Module Name**: Media, Iframe |
| Add media content in a frame using audio, video, iframe elements to the Home page of IEKart's Shopping application. |
| Exercise-3 | 3.a | **Course Name**: Javascript |
| **Module Name**: Type of Identifiers |
| Write a JavaScript program to find the area of a circle using radius (var and let - reassign and observe the difference with var and let) and PI (const) |
| 3.b | **Course Name**: Javascript |
| **Module Name**: Primitive and Non Primitive Data Types |
| Write JavaScript code to display the movie details such as movie name, starring, language, and ratings. Initialize the variables with values of appropriate types. Use template literals wherever necessary. |
| 3.c | **Course Name**: Javascript |
| **Module Name**: Operators and Types of Operators |
| Write JavaScript code to book movie tickets online and calculate the total price, considering the number of tickets and price per ticket as Rs. 150. Also, apply a festive season discount of 10% and calculate the discounted amount. |
| 3.d | **Course Name**: Javascript |
| **Module Name**: Types of Statements, Non - Conditional Statements, Types of Conditional Statements, if Statements, switch Statements |
| Write a JavaScript code to book movie tickets online and calculate the total price based on the 3 conditions: (a) If seats to be booked are not more than 2, the cost per ticket remains Rs. 150. (b) If seats are 6 or more, booking is not allowed. (c) If se |
| 3.e | **Course Name**: Javascript |
| **Module Name**: Types of Loops |
| Write a JavaScript code to book movie tickets online and calculate the total price based on the 3 conditions: (a) If seats to be booked are not more than 2, the cost per ticket remains Rs. 150. (b) If seats are 6 or more, booking is not allowed. (c) If |
| Exercise-4 | 4.a | **Course Name**: Javascript |
| **Module Name**: Types of Functions, Declaring and Invoking Function, Arrow Function, Function Parameters, Nested Function, Built-in Functions, Variable Scope in Functions |
| Write a JavaScript code to book movie tickets online and calculate the total price based on the 3 conditions: (a) If seats to be booked are not more than 2, the cost per ticket remains Rs. 150. (b) If seats are 6 or more, booking is not allowed. (c) If |
| 4.b | **Course Name**: Javascript |
| **Module Name**: Working With Classes, Creating and Inheriting Classes |
| Create an Employee class extending from a base class Person. Hints: (i) Create a class Person with name and age as attributes. (ii) Add a constructor to initialize the values (iii) Create a class Employee extending Person with additional attributes role |
| 4.c | **Course Name**: Javascript |
| **Module Name**: In-built Events and Handlers |
| Write a JavaScript code to book movie tickets online and calculate the total price based on the 3 conditions: (a) If seats to be booked are not more than 2, the cost per ticket remains Rs. 150. (b) If seats are 6 or more, booking is not allowed. (c) If se |
| 4.d | **Course Name**: Javascript |
| **Module Name**: Working with Objects, Types of Objects, Creating Objects, Combining and cloning Objects using Spread operator, Destructuring Objects, Browser Object Model, Document Object Model |
| If a user clicks on the given link, they should see an empty cone, a different heading, and a different message and a different background color. If user clicks again, they should see a re-filled cone, a different heading, a different message, and a diffe |
| Exercise-5 | 5.a | **Course Name**: Javascript |
| **Module Name**: Creating Arrays, Destructuring Arrays, Accessing Arrays, Array Methods |
| Create an array of objects having movie details. The object should include the movie name, starring, language, and ratings. Render the details of movies on the page using the array. |
| 5.b | **Course Name**: Javascript |
| **Module Name**: Introduction to Asynchronous Programming, Callbacks, Promises, Async and Await, Executing Network Requests using Fetch API |
| Simulate a periodic stock price change and display on the console. Hints: (i) Create a method which returns a random number - use Math.random, floor and other methods to return a rounded value. (ii) Invoke the method for every three seconds and stop when |
| 5.c | **Course Name**: Javascript |
| **Module Name**: Creating Modules, Consuming Modules |
| Validate the user by creating a login module. Hints: (i) Create a file login.js with a User class. (ii) Create a validate method with username and password as arguments. (iii) If the username and password are equal it will return "Login Successful" else w |
| Exercise-6 | 6.a | **Course Name**: Node.js |
| **Module Name**: How to use Node.js |
| Verify how to execute different functions successfully in the Node.js platform. |
| 6.b | **Course Name**: Node.js |
| **Module Name**: Create a web server in Node.js |
| Write a program to show the workflow of JavaScript code executable by creating web server in Node.js. |
| 6.c | **Course Name**: Node.js |
| **Module Name**: Modular programming in Node.js |
| Write a Node.js module to show the workflow of Modularization of Node application. |
| 6.d | **Course Name**: Node.js |
| **Module Name**: Restarting Node Application |
| Write a program to show the workflow of restarting a Node application. |
| 6.e | **Course Name**: Node.js |
| **Module Name**: File Operations |
| Create a text file src.txt and add the following data to it. Mongo, Express, Angular, Node. |
| Exercise-7 | 7.a | **Course Name**: Express.js |
| **Module Name**: Defining a route, Handling Routes, Route Parameters, Query Parameters |
| Implement routing for the AdventureTrails application by embedding the necessary code in the routes/route.js file. |
| 7.b | **Course Name**: Express.js |
| **Module Name**: How Middleware works, Chaining of Middlewares, Types of Middlewares |
| In myNotes application: (i) we want to handle POST submissions. (ii) display customized error messages. (iii) perform logging. |
| 7.c | **Course Name**: Express.js |
| **Module Name**: Connecting to MongoDB with Mongoose, Validation Types and Defaults |
| Write a Mongoose schema to connect with MongoDB. |
| 7.d | **Course Name**: Express.js |
| **Module Name**: Models |
| Write a program to wrap the Schema into a Model object. |
| Exercise-8 | 8.a | **Course Name**: Express.js |
| **Module Name**: CRUD Operations |
| Write a program to perform various CRUD (Create-Read-Update-Delete) operations using Mongoose library functions. |
| 8.b | **Course Name**: Express.js |
| **Module Name**: API Development |
| In the myNotes application, include APIs based on the requirements provided. (i) API should fetch the details of the notes based on a notesID which is provided in the URL. Test URL - http://localhost:3000/notes/7555 (ii) API should update the details bas |
| 8.c | **Course Name**: Express.js |
| **Module Name**: Why Session management, Cookies |
| Write a program to explain session management using cookies. |
| 8.d | **Course Name**: Express.js |
| **Module Name**: Sessions |
| Write a program to explain session management using sessions. |
| 8.e | **Course Name**: Express.js |
| **Module Name**: Why and What Security, Helmet Middleware |
| Implement security features in myNotes application |
| Exercise-9 | 9.a | **Course Name**: Typescript |
| **Module Name**: Basics of TypeScript |
| On the page, display the price of the mobile-based in three different colors. Instead of using the number in our code, represent them by string values like GoldPlatinum, PinkGold, SilverTitanium. |
| 9.b | **Course Name**: Typescript |
| **Module Name**: Function |
| Define an arrow function inside the event handler to filter the product array with the selected product object using the productId received by the function. Pass the selected product object to the next screen. |
| 9.c | **Course Name**: Typescript |
| **Module Name**: Parameter Types and Return Types |
| Consider that developer needs to declare a function - getMobileByVendor which accepts string as input parameter and returns the list of mobiles. |
| 9.d | **Course Name**: Typescript |
| **Module Name**: Arrow Function |
| Consider that developer needs to declare a manufacturer's array holding 4 objects with id and price as a parameter and needs to implement an arrow function - myfunction to populate the id parameter of manufacturers array whose price is greater than or equ |
| 9.e | **Course Name**: Typescript |
| **Module Name**: Optional and Default Parameters |
| Declare a function - getMobileByManufacturer with two parameters namely manufacturer and id, where manufacturer value should passed as Samsung and id parameter should be optional while invoking the function, if id is passed as 101 then this function shoul |
| Exercise-10 | 10.a | **Course Name**: Typescript |
| **Module Name**: Rest Parameter |
| Implement business logic for adding multiple Product values into a cart variable which is type of string array. |
| 10.b | **Course Name**: Typescript |
| **Module Name**: Creating an Interface |
| Declare an interface named - Product with two properties like productId and productName with a number and string datatype and need to implement logic to populate the Product details. |
| 10.c | **Course Name**: Typescript |
| **Module Name**: Duck Typing |
| Declare an interface named - Product with two properties like productId and productName with the number and string datatype and need to implement logic to populate the Product details. |
| 10.d | **Course Name**: Typescript |
| **Module Name**: Function Types |
| Declare an interface with function type and access its value. |
| Exercise-11 | 11.a | **Course Name**: Typescript |
| **Module Name**: Extending Interfaces |
| Declare a productList interface which extends properties from two other declared interfaces like Category,Product as well as implementation to create a variable of this interface type. |
| 11.b | **Course Name**: Typescript |
| **Module Name**: Classes |
| Consider the Mobile Cart application, Create objects of the Product class and place them into the productlist array. |
| 11.c | **Course Name**: Typescript |
| **Module Name**: Constructor |
| Declare a class named - Product with the below-mentioned declarations: (i) productId as number property (ii) Constructor to initialize this value (iii) getProductId method to return the message "Product id is <>". |
| 11.d | **Course Name**: Typescript |
| **Module Name**: Access Modifiers |
| Create a Product class with 4 properties namely productId, productName, productPrice, productCategory with private, public, static, and protected access modifiers and accessing them through Gadget class and its methods. |
| Exercise-12 | 12.a | **Course Name**: Typescript |
| **Module Name**: Properties and Methods |
| Create a Product class with 4 properties namely productId and methods to setProductId() and getProductId(). |
| 12.b | **Course Name**: Typescript |
| **Module Name**: Creating and using Namespaces |
| Create a namespace called ProductUtility and place the Product class definition in it. Import the Product class inside productlist file and use it. |
| 12.c | **Course Name**: Typescript |
| **Module Name**: Creating and using Modules |
| Consider the Mobile Cart application which is designed as part of the functions in a module to calculate the total price of the product using the quantity and price values and assign it to a totalPrice variable. |
| 12.d | **Course Name**: Typescript |
| **Module Name**: What is Generics, What are Type Parameters, Generic Functions, Generic Constraints |
| Create a generic array and function to sort numbers as well as string values. |

**NAME BOARD**

**Size- (1\*1) Count- 1**

|  |  |
| --- | --- |
| LAB ASSISTANT | **U.SASI KIRAN**  **LAB ASSISTANT** |