**TEKISKY PVT.LTD**.

**IT industry and its trends:**

IT industry is a dynamic sector that is focused on leveraging technology for innovation and efficiency.

1. Product-based companies:

Product Based companies are the ones that produce high-end products in order to satisfy their customers’ requirements. These companies are focused on introducing or producing products that have a high market value.

Ex, google, Microsoft and oracle which develop and sell hardware, software, and cloud services

1. Service-based companies:

Service-based companies are the ones that do not have a particular product but their primary business is to provide services. The services they provide are intangible and can be provided in the terms of expertise, amenities, or skills. These companies generally provide less salary compared to product-based companies. Some of the popular service-based companies are TCS, Infosys, Wipro, Tech Mahindra, etc.

1. Cloud providers:

It is a third-party company that provides scalable computing resources that businesses can access on demand over a network, including cloud-based compute, storage, platform, and application services.

Ex,AWS, MS Azure and google cloud offer scalable cloud computing solution.

**Roles in IT sector:**

1. Board members(CEO, Director):

They are involved in high level decision making and governance , provide stratergic direction oversight company growth and operation.

1. Delivery manager:

Focus on successful delivery of software project , co-ordinates between teams , clients expectations.

1. Manager:

Overseas the execution of projects, manages resources and ensures goals are achieved with budget and schedule.

1. Consultant:

Provides expert advice and solutions on technology strategies.

1. Team lead(TL):

Manages team of developers, co-ordinates the tasks.

1. Technical architect:

Designs the structure of software systems.

1. Senior software engineer:

Manage the development of complex software projects, designs systems architecture, guide junior engineer.

1. Junior software engineer:

Test the software application, resolve bugs and errors.

**SOFTWARE DEVELOPMENT LIFE CYCLE:**

The Software Development Life Cycle (SDLC) is a structured process that enables the production of high-quality, low-cost software, in the shortest possible production time. The goal of the SDLC is to produce superior software that meets and exceeds all customer expectations and demands. The SDLC defines and outlines a detailed plan with stages, or phases, that each encompass their own process and deliverables.

1. Planning
2. Requirement
3. Design and prototyping
4. Software development
5. Testing
6. Deployment
7. Operation and maintenance

**Methodologies:**

1. Waterfall:

represents the oldest, simplest, and most structured methodology. Each phase depends on the outcome of the previous phase, and all phases run sequentially. This model provides discipline and gives a tangible output at the end of each phase. However, this model doesn’t work well when flexibility is a requirement. There is little room for change once a phase is deemed complete, as changes can affect the cost, delivery time, and quality of the software.

1. Agile:

The agile methodology produces ongoing release cycles, each featuring small, incremental changes from the previous release. At each iteration, the product is tested. The agile model helps teams identify and address small issues in projects before they evolve into more significant problems. Teams can also engage business stakeholders and get their feedback throughout the development process.

**JOB OPPORTUNITIES:**

* AI:

Artificial intelligence (AI) is technology that enables computers and machines to simulate human learning, comprehension, problem solving, decision making, creativity and auton

**Automation of repetitive**

**More and faster insight from data.**

**Enhanced decision-making.**

**Fewer human errors.**

**24x7 availability.**

**Reduced physical risks.**

* Machine learning:

 involves creating [models](https://www.ibm.com/topics/ai-model) by training an algorithm to make predictions or decisions based on data. It encompasses a broad range of techniques that enable computers to learn from and make inferences based on data without being explicitly programmed for specific tasks.

* Data science:

 uses algorithms, procedures, and processes to examine large amounts of data to uncover hidden patterns, generate insights, and direct decision-making. Data scientists use advanced machine learning algorithms to sort through, organise and learn from structured and unstructured data to create prediction models.

* IOT(internet of things):  
   is a network of interrelated devices that connect and exchange data with other IoT devices and the cloud. IoT devices are typically embedded with technology such as sensors and software and can include mechanical and digital machines and consumer objects.

Ex, android watch/smart watches, fingerprint sensor.

* Automation:

Less human interaction, increased productivity.

(conveyer belt)

* Cloud computing:

storing and accessing the data and programs on remote servers that are hosted on the internet instead of the computer’s hard drive or local server. Cloud computing is also referred to as Internet-based computing, it is a technology where the resource is provided as a service through the Internet to the user. The data that is stored can be files, images, documents, or any other storable document.

**Basic Tools:**

Python: For data science and AI.

TensorFlow: For machine learning.

AWS/Azure: For cloud services.

**TECHNOLOGIES:**

IoT: Enables communication between interconnected devices. Automation: Uses technology to perform repetitive tasks with minimal human intervention.

Cloud: Offers flexible and scalable computing services over the internet. Full Stack: Combines both client-side and server-side development for comprehensive application creation.

AI Tools Examples: Keras, PyTorch, and OpenAI GPT-4.

**FRONT END TECHNOLOGIES:**

React Angular VUE, HTML, Javascript

**FlexBox Game:**

It is a one-dimensional layout method for arranging items in rows or columns. Items flex (expand) to fill additional space or shrink to fit into smaller spaces.

**The froggy game:**

To move right by using the **justify-content** property, which aligns items horizontally and accepts the following values:

* **flex-start**: Items align to the left side of the container.
* **flex-end**: Items align to the right side of the container.
* **center**: Items align at the center of the container.
* **space-between**: Items display with equal spacing between them.
* **space-around**: Items display with equal spacing around them.

To move frog to right side ;

#pond{

Display: flex;

Justify-content:flex-end;

}

**usealign-items** to help the frogs get to the bottom of the pond. This CSS property aligns items vertically and accepts the following values:

* **flex-start**: Items align to the top of the container.
* **flex-end**: Items align to the bottom of the container.
* **center**: Items align at the vertical center of the container.
* **baseline**: Items display at the baseline of the container.
* **stretch**: Items are stretched to fit the container.

To move frog to the bottom:

#pond{

Display:flex;

Align-items:flex-end;

}

To move frog to the center:

#pond{

Display:flex;

Justify-content:center;

Align-items:flex-end;

}

The frogs need to get in the same order as their lilypads using **flex-direction**. This CSS property defines the direction items are placed in the container, and accepts the following values:

* **row**: Items are placed the same as the text direction.
* **row-reverse**: Items are placed opposite to the text direction.
* **column**: Items are placed top to bottom.
* **column-reverse**: Items are placed bottom to top.

To change the direction :

#pond{

Display: flex;

Flex-direction: row-reverse;

}

**BACKEND TECHNOLOGIES:**

A **back-end framework** is used to create server-side web architectures stably and efficiently. **Backend technologies** focus on improving the hidden aspects of the websites and are very important for the proper execution of any **web application**. It enables the smooth creation and maintenance of a website.

Backend development focuses on managing the **server-side functionalities** and implementation with tasks like **designing APIs**, dealing with various components, etc.

## ****Node.js:****

Node.js is an open-source, cross-platform **JavaScript framework** that is used to build server-side and networking applications. [Node.js](https://www.geeksforgeeks.org/introduction-to-node-js/)is built using **Google’s V8 JavaScript engine**. Node.js provides various JavaScript modules and integration with various third-party libraries and APIs to develop web applications easily and efficiently.

The Node.js framework makes it easy for developers to write command-line tools and server scripts with the help of JavaScript.

High performance, scalability, rich community, reduced loading time.

* PYTHON:

Python is a powerful, versatile, and popular backend language used for web development. It is one of the most popular backend technologies in the list. has a simple syntax which makes it easy to learn and use. Python allows developers to quickly develop applications without having to write complex code. It also supports several frameworks, such as Django, Flask, and Pyramid, that make web development easier. USED FOR API CALLING

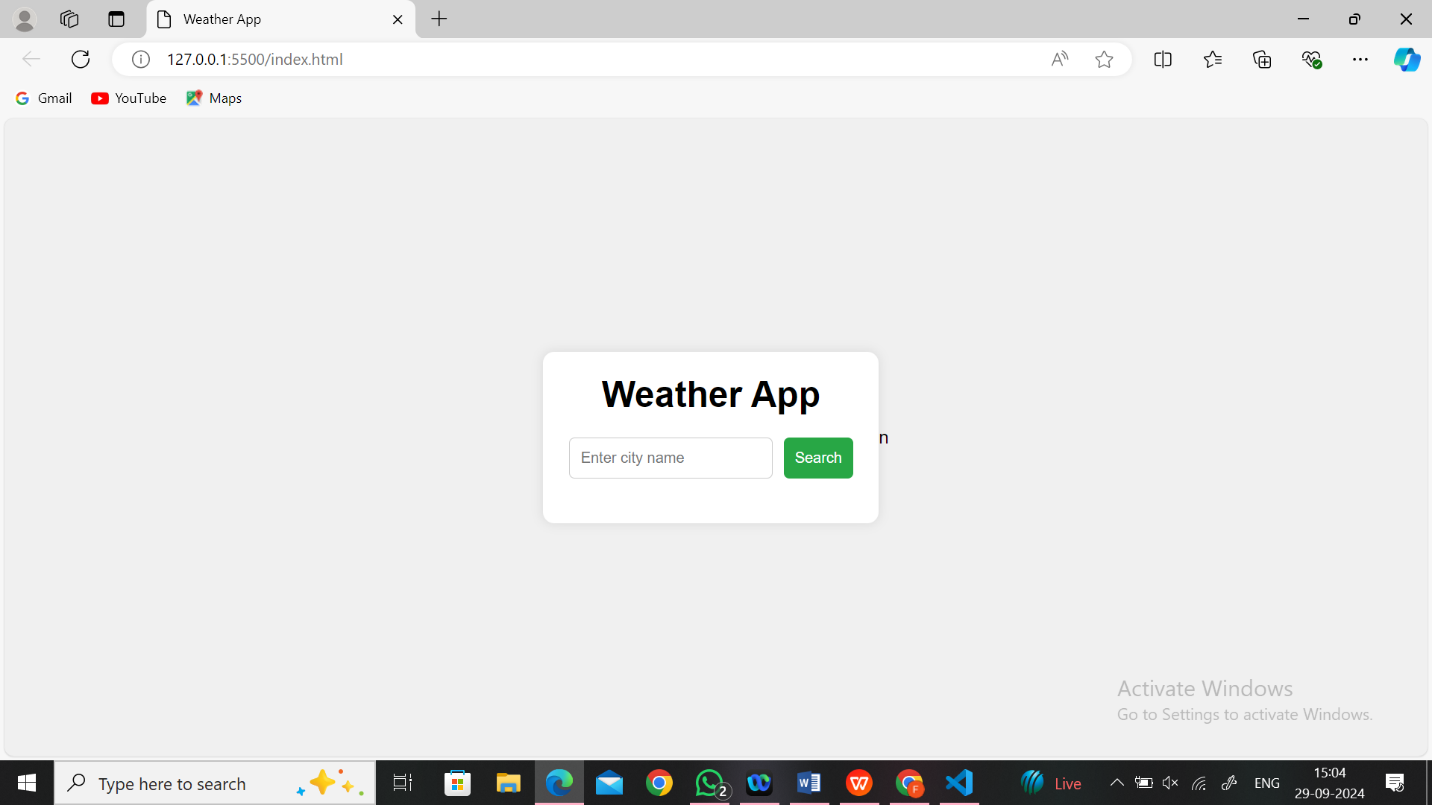
* MONGO DB:

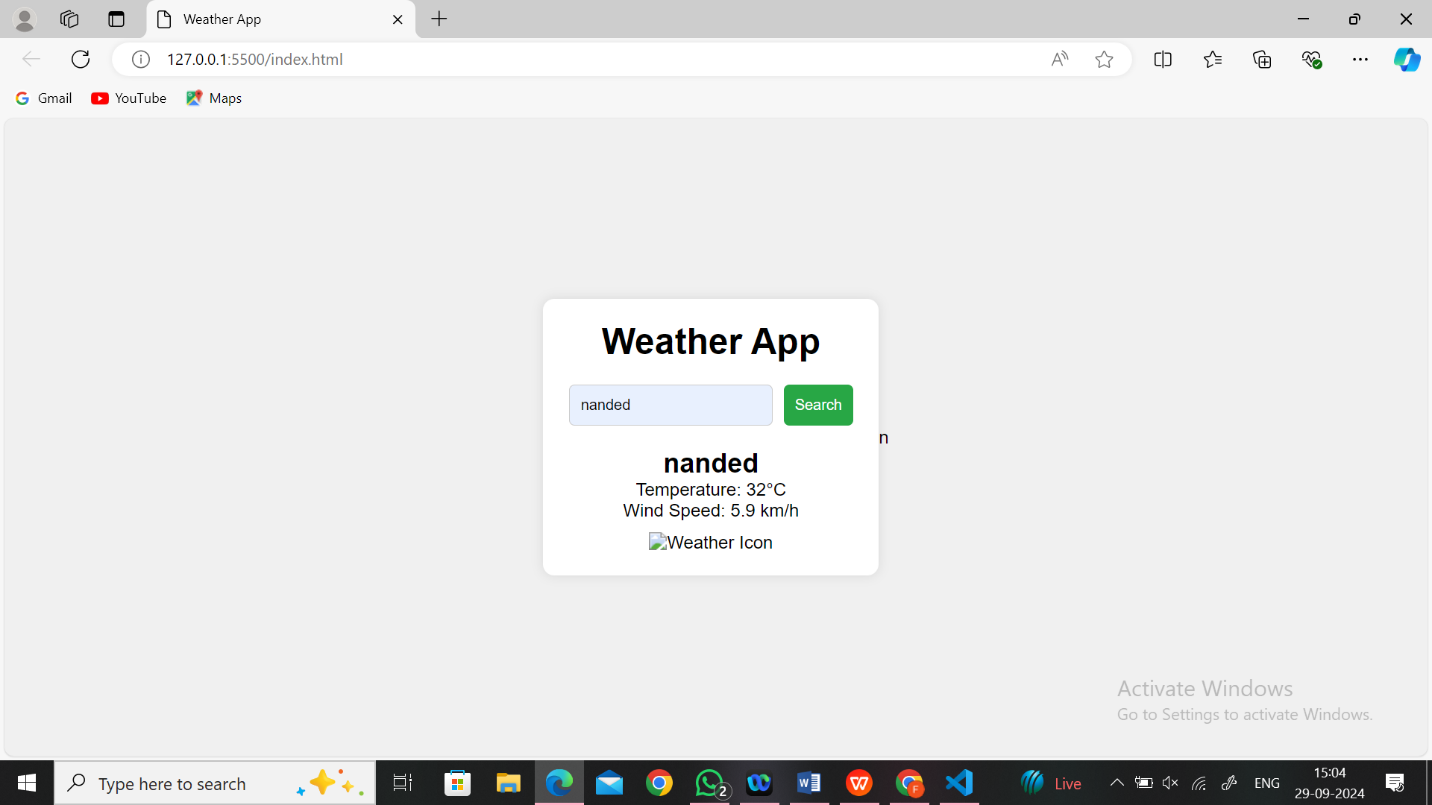
MongoDB is an open-source NoSQL database management system designed to store data in a document-oriented format. It is one of the most popular databases used for web development today. MongoDB offers scalability, high performance, easy querying of data, and integration with various languages such as JavaScript, Python, C++, Java, and more.

* EXPRESS:

LIGHT-WEIGHT FRAMEWORK FOR USER SERVERS.

**DEVELOPING WEATHER APP USINF VS CODE:**

****

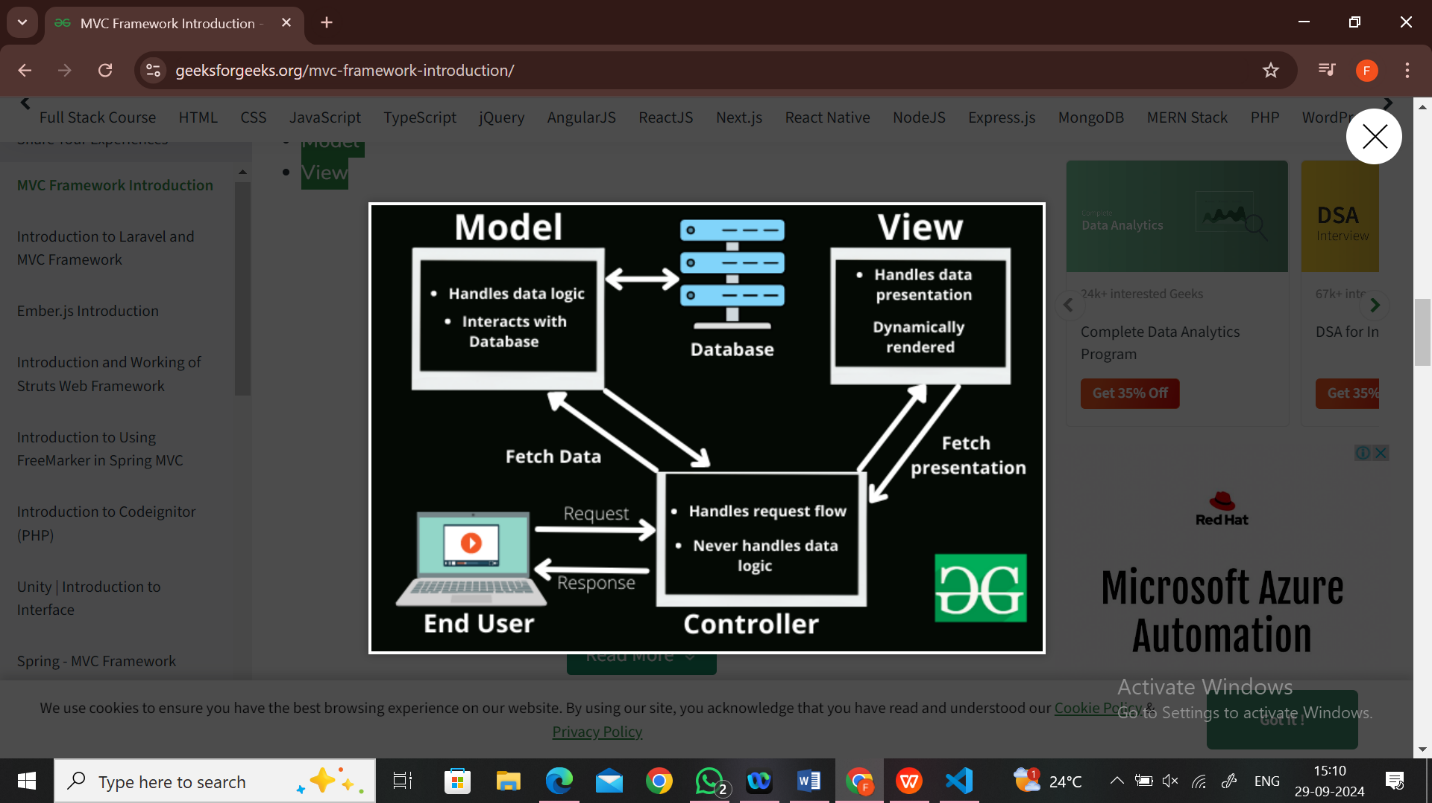


## MVC ARCHITECTURE:

## The ****Model-View-Controller (MVC)**** framework is an architectural/design pattern that separates an application into three main logical components ****Model****, ****View****, and ****Controller****. Each architectural component is built to handle specific development aspects of an application. It isolates the business logic and presentation layer from each other. It was traditionally used for desktop ****graphical user interfaces (GUIs)****. Nowadays, MVC is one of the most frequently used industry-standard web development frameworks to create scalable and extensible projects. It is also used for designing mobile apps.

The MVC framework includes the following 3 components:

* Controller
* Model
* View



### **Controller:**

The controller is the component that enables the interconnection between the views and the model so it acts as an intermediary. The controller doesn’t have to worry about handling data logic, it just tells the model what to do.

### **View:**

The **View**component is used for all the UI logic of the application. It generates a user interface for the user. Views are created by the data which is collected by the model component but these data aren’t taken directly but through the controller. It only interacts with the controller

### **Model:**

The **Model**component corresponds to all the data-related logic that the user works with. This can represent either the data that is being transferred between the View and Controller components or any other business logic-related data. It can add or retrieve data from the database. It responds to the controller’s request because the controller can’t interact with the database by itself.

**DREAM WEBIES PVT.LTD**

**Introduction to UI/UX Design:**

UX design refers to the term “[user experience design](https://careerfoundry.com/en/blog/ux-design/what-is-user-experience-ux-design-everything-you-need-to-know-to-get-started/)”, while UI stands for “[user interface design](https://careerfoundry.com/en/blog/ui-design/what-is-ui-design-guide/)”. Both elements are crucial to a product and work closely together. But despite their professional relationship, the roles themselves are quite different, referring to very different aspects of the product development process and the design discipline.

A UX designer thinks about how the experience makes the user feel and how easy it is for the user to accomplish their desired tasks. They also observe and conduct [task analyses](https://careerfoundry.com/en/blog/ux-design/task-analysis-ux/) to see how users actually complete tasks in a user flow.

The ultimate purpose of UX design is to create easy, efficient, relevant, and all-around pleasant experiences for the user.

UX design is NOT about visuals; it focuses on the overall feel of the experience.

Unlike UX, user interface design is a strictly digital term.

[A user interface](https://careerfoundry.com/en/blog/ui-design/what-is-a-user-interface/) is the point of interaction between the user and a digital device or product—like the touchscreen on your smartphone or the touchpad you use to select what kind of coffee you want from the coffee machine.

In relation to websites and apps, UI design considers the look, feel, and interactivity of the product. It’s all about making sure that the user interface of a product is as intuitive as possible, and that means carefully considering each and every visual, interactive element the user might encounter.

A UI designer will think about [icons and buttons](https://careerfoundry.com/en/blog/ui-design/icon-design-process/), [typography](https://careerfoundry.com/en/blog/ui-design/beginners-guide-to-typography/) and [color schemes](https://careerfoundry.com/en/blog/ui-design/introduction-to-color-theory-and-color-palettes/), spacing, imagery, and responsive design

Key principles of good design: consistency, simplicity, and user centered design

Tools like Figma or Adobe XD for creating wireframes and mockups

**Introduction to MERN Stack:**

MERN is a pre-built technology stack based on JavaScript technologies. MERN stands for **M**ongoDB, **E**xpress, **R**eact, and **N**ode, after the four key technologies that make up the stack.

* MongoDB — document database
* Express(.js) — Node.js web framework
* React(.js) — a client-side JavaScript framework
* Node(.js) — the premier JavaScript web server (runtime)

Express and Node make up the middle (application) tier. Express.js is a server-side web framework, and Node.js is the popular and powerful JavaScript server platform.

REACT.JS front end:

React lets you build complex interfaces through simple components, connect them to data on your back-end server, and render them as HTML.It has all the features you'd expect from a modern web framework: great support for forms, error handling, events, lists, and more.

Express.js and Node.js server tier:

Express.js bills itself as a “fast, unopinionated, minimalist web framework for Node.js,” and that is indeed exactly what it is. Express.js has powerful models for URL routing (matching an incoming URL with a server function), and handling HTTP requests and responses.

### MongoDB database tier**:** If your application stores any data (user profiles, content, comments, uploads, events, etc.), then you're going to want a database that's just as easy to work with as React, [Express](https://www.mongodb.com/resources/products/compatibilities/express), and Node.js.That's where MongoDB comes in: JSON documents created in your React.js front end can be sent to the Express.js server, where they can be processed and (assuming they're valid) stored directly in MongoDB for later retrieval.