PROJECT REPORT

Submitted to

DEPARTMENT OF COMPUTER SCIENCE

VIVEKANAND EDUCATION SOCIETY'S

COLLEGE OF ARTS, SCIENCE AND COMMERCE,

SINDHI SOCIETY, CHEMBUR, MUMBAI 400071.

MindGen AI

"Generate Brilliance, Mind by Mind"

For Partial Fulfilment for Degree of

Bachelor of Science(Computer Science)

Academic Year(2023-24)

COORDINATOR OF DEPARTMENT	COLLEGE GUIDE
Mr. Kamlakar Bhopatkar	**Mr. Kamlakar Bhopatkar**
	Ms. Priya Daniel

SUBMITTED BY

Krishna R. Kushwaha

VIVEKANAND EDUCATION SOCIETY'S
COLLEGE OF ARTS, SCIENCE AND COMMERCE,
SINDHI SOCIETY, CHEMBUR, MUMBAI 400071.
NAAC Re-Accredited 'A' Grade

CERTIFICATE

This is to certify that **Mr. Krishna R. Kushwaha** of T.Y.B.Sc(Computer Science) affiliated to University of Mumbai has successfully completed a project work entitled.

MindGen AI
"Generate Brilliance, Mind by Mind"

As partial fulfilment of the requirement for the degree of B.Sc.(Computer Science) for the Academic Year 2023-24.

COORDINATOR OF DEPARTMENT

Mr. Kamlakar Bhopatkar

COLLEGE GUIDE

Mr. Kamlakar Bhopatkar
Ms. Priya Daniel

Date:

Examiner:

College Seal

Acknowledgement:

With great pleasure, I present my project, **'MindGen AI**,' and extend sincere gratitudeto all those whose invaluable contributions have been pivotal to its success.

This project report stands as a testament to collective efforts, and I express heartfelt thanks to **V.E.S. College** and our esteemed Principal, **Dr. Anita Kanwar**, for their unwavering support.

A special acknowledgment is reserved for our dedicated Project Guides, **Mr. Kamlakar Bhopatkar** and **Ms. Priya Daniel**, whose timely and invaluable guidance played a crucial role in each phase of the project's development.

The synergy among the project team and the mentorship provided were integral to overcoming challenges and achieving the milestones outlined in the 'MindGen AI,'

This endeavor has been a true collaboration, and I extend my gratitude to every individual who has contributed to its success. Your collective efforts have not only shaped this project but have also enriched the overall learning experience.

In closing, I express my deepest appreciation to all involved in making 'MindGen AI,' a reality. Your support and guidance have been indispensable, and I am genuinely thankful for the privilege of undertaking this project within such a supportive community.

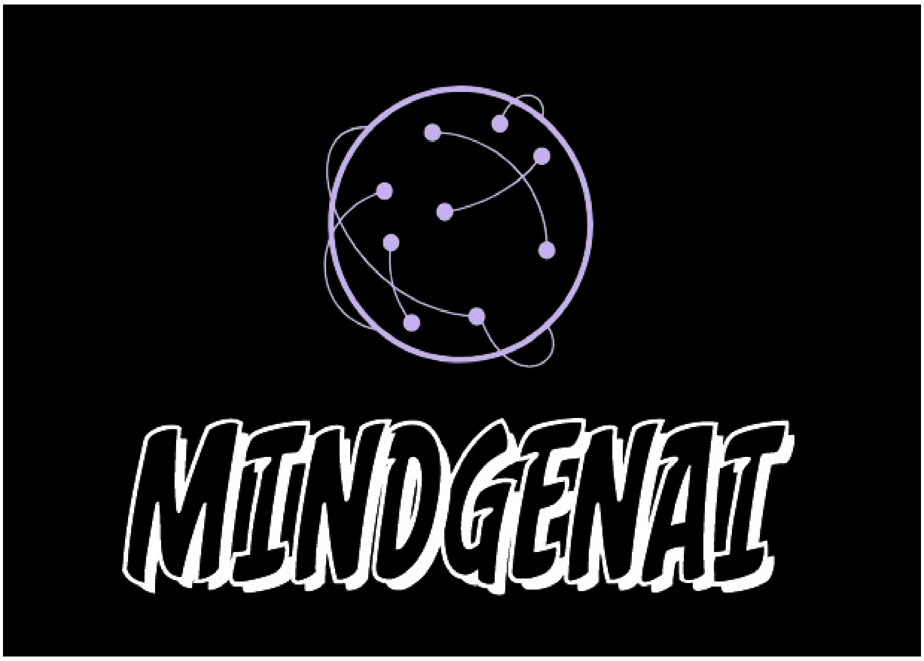
As I reflect on this journey, I am inspired by the collective commitment to excellence and the pursuit of knowledge. This experience has not only honed my technical skills but has also instilled in me a deep sense of gratitude for the collaborative spirit that defines our academic community.

I look forward to applying the knowledge gained from this project as I embark on future endeavours, and I am confident that the lessons learned will serve as a solid foundation for my continued growth in the field.

INDEX

MindGen AI

"Generate Brilliance, Mind by Mind"



Preliminary Investigation

Organizational overview:

MindGen AI is a cutting-edge educational platform revolutionizing the way we learn by seamlessly integrating artificial intelligence into course generation. Our mission is to empower learners with personalized and dynamic educational experiences, fostering a deep understanding of diverse subjects. At MindGen AI, we believe in the transformative power of knowledge and aim to make learning an inspiring and accessible journey for everyone.

Description of Existing System:

Embarking on an educational journey, a prominent online learning platform, unveils a world of opportunities in technology-focused fields. However, as we explore the vast landscape of possibilities, it becomes imperative to shine a light on the complex aspects acknowledging potential drawbacks that might influence one's decision-making process. By understanding and contemplating these limitations, learners can navigate their educational path with a comprehensive perspective, ensuring that the chosen platform aligns seamlessly with their individual needs and aspirations. Let us delve into a thoughtful examination of the drawbacks associated with Udacity, enhancing our understanding of the platform's dynamics.

Limitations of Existing System:

Taking an online course opens you a world of opportunities for professional growth and skill development. There are several barriers that students should be aware of before embarking on their educational journey. It is essential to illuminate certain limitations that learners may encounter on their educational voyage.

Here is a list of the limitations.

1. **Cost:**
   Course programs often come with a price tag, which might be a significant consideration for individuals on a tight budget. While they offer value, the financial commitment could be a limitation for some learner.

2. **Limited Course Diversity:**
   The main areas of interest are technology and associated industries. Compared to platforms that offer a larger range of disciplines, you might find the choices to be restricted if you are seeking for a broad range of courses outside of technology.

3. **Self-Directed Learning Challenges:**
   The flexibility of self-paced learning can be a double-edged sword. **Some individuals may struggle with the autonomy,** needing more structure and guidance that a traditional classroom setting provides.

4. **Time Commitment:**
   The flexibility of self-paced learning can be a double-edged sword, as it requires a significant time commitment and self-discipline. Balancing coursework with other commitments may be challenging for those with busy schedules or fluctuating workloads.

Description Proposed system:

MindGen AI is at the forefront of educational innovation, leveraging the power of artificial intelligence to transform traditional learning paradigms. With a passion for knowledge and a commitment to excellence, we are redefining the educational landscape by providing learners with an intelligent and dynamic platform that adapts to their unique needs.

MindGen AI is built on the philosophy that learning is a dynamic and personal experience. Our AI algorithms analyze individual learning patterns, preferences, and strengths to craft courses that not only deliver information but also inspire a deeper understanding and retention of knowledge

Powered by cutting-edge technology, MindGen AI stands at the intersection of artificial intelligence, machine learning, and educational expertise. Our commitment to technological excellence ensures that learners receive the most advanced and effective learning experiences

Advantages of Proposed system:

1. **AI-Powered Course Generation:**
   MindGen AI utilizes advanced AI algorithms to create personalized and dynamic course content tailored to individual learning needs.
2. **Seamless Integration:**
   Our platform seamlessly integrates with various APIs, including OpenAI, YouTube, and Stripe, providing a comprehensive and feature-rich learning experience.

3. **User-Centric Design**
   MindGen AI prioritizes user experience, offering an intuitive interface for easy navigation and a delightful learning journey.

Modules:

This module aims to introduce learners, educators, and stakeholders to the exciting world of MindGen AI, providing a foundational understanding of its core values and the transformative potential it brings to the field of education.

This module focuses on a detailed evaluation of the existing system, providing insights into the challenges learners face on platforms like Udacity. It lays the groundwork for understanding the need for a more advanced and personalized approach, as proposed by MindGen AI.

1. **Login Module:**
   A login module is a software component or module that oversees authenticating users and granting them access to a certain system or application. The primary function of a login module is to guarantee that only authorized users have access to restricted services or information.

2. **Course Form:**
   The course form is your go-to document for a quick overview of what a course entails. It covers the basics course title, duration, and instructor while also diving into crucial details like learning objectives, prerequisites, and assessment methods. Consider it your roadmap to a successful learning experience, offering all the essential information in a concise format.

3. **Chapters Endpoints:**
   Chapters Endpoints streamline your navigation through course content. These endpoints mark the conclusion of specific sections, providing a clear transition to the next phase of your learning journey. Think of them as signposts guiding you seamlessly through the chapters, ensuring a structured and organized learning experience.

Technologies Used:

1. **HTML**
   HTML is the standard markup language for creating webpages.

2. **CSS**
   CSS beautifies websites by styling HTML elements, dictating their appearance, and ensuring a seamless user experience.

3. **Typescript:**
   TypeScript, a superset of JavaScript, introduces static typing for early error detection, It provides a structured layer, enhancing development efficiency and code reliability.

4. **Node JS**
   Node js is simple to use as a server-side proxy since it can manage several connections at once in a nonblocking fashion. It is helpful for gathering data from several source points or proxying various services with diverse response times.

5. **Next.js**
   Next.js is a React framework that simplifies and accelerates web development, providing a smooth experience for building modern, server-rendered React applications.

6. **Shadcn**
   Shadcn UI is a collection of reusable components that can be easily installed and used in apps for customization

7. **Prisma**
   Prisma's main goal is to make application developers more productive when working with databases.

8. **MongoDB**
   MongoDB, a NoSQL database, redefines data storage with flexible JSON-like documents. Its scalability and speed make it a top choice for dynamic data management.

SYSTEM ANALYSIS

Use Case:

Use case diagrams are used to depict the context of the system to be built and the functionality provided by that system. They depict who (or what) interacts with the system. They show what the outside world wants the system to do

ELEMENTS OF USE CASE DIAGRAM:

Actors:

Any entity (or entities) that play specific roles in a particular system are portrayed by actors. The many roles that the actor plays correspond to the users' actual job functions inside a particular system. In a use case diagram, an actor engages in interaction with a use case. All of the stakeholders, who are the target users of the app, are involved. It makes no sense to represent an entity as an actor if it has no impact on the functionality you are simulating. In a use case diagram that is portrayed "outside" the system boundaries, an actor is represented by a stick figure.

Use Cases:

A use case is a graphic depiction of a specific piece of business functionality in a system. You should make a list of the discrete business functions in the form of simple scenarios as the first step in determining use cases. These business operations may all be categorised as possible use cases. In a use case diagram, a use case is represented by an ellipse.

System Boundary:

A system boundary defines the scope of what a system will be. The system boundary is shown as a rectangle spanning all the use cases in the system.

Relationships: The following relationships can be established among use cases:

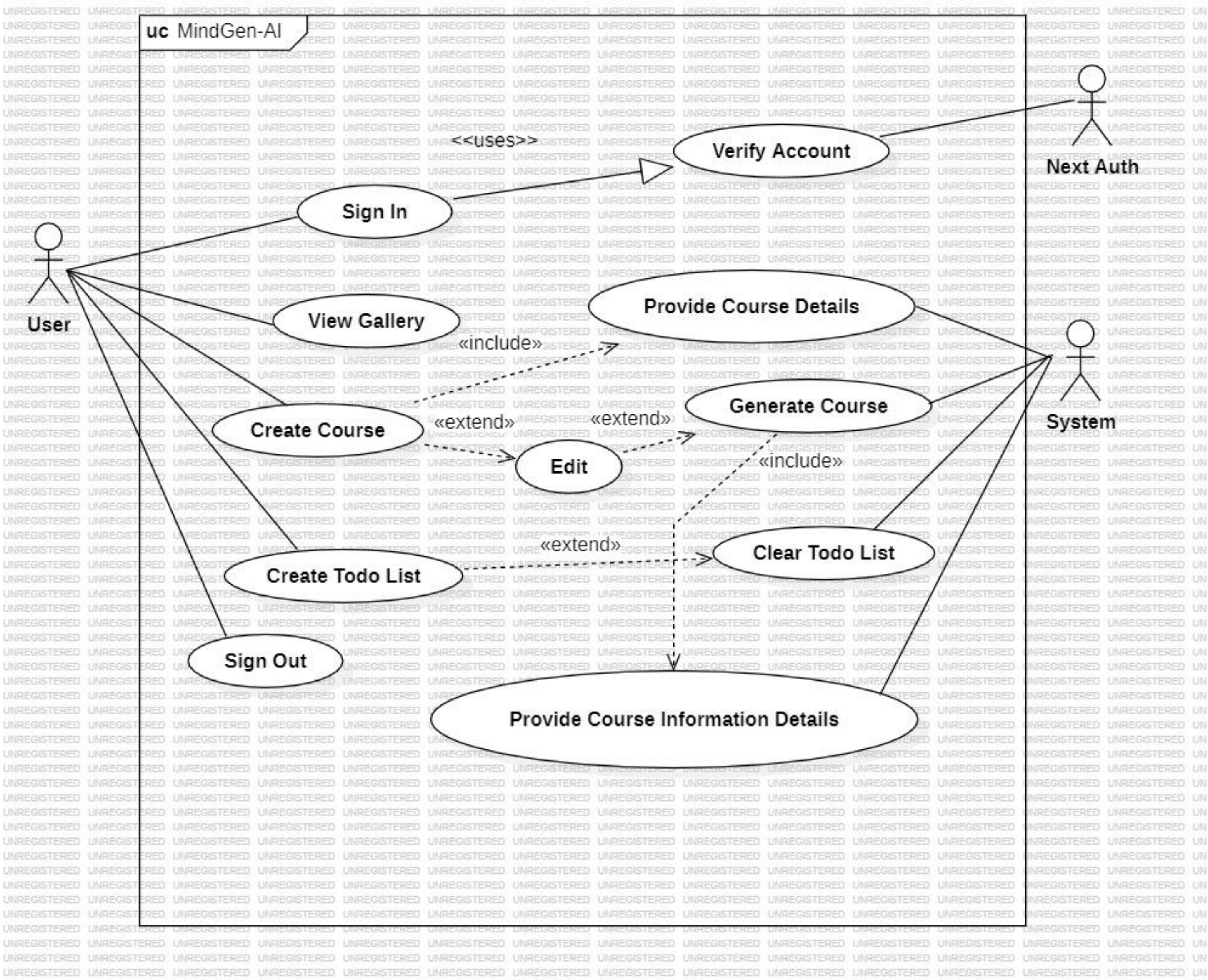
Extends:

A use case may extend another. This relationship indicates that the behavior of the extension use case may be inserted in the extended use case under some conditions. It is used for showing optional use cases after a particular use case. The notation is a dashed arrow from the optional case to the base use case, with the label “«extend»"

Includes:

A use case may include another. Include is a Directed Relationship between two use cases, implying that the behavior of the included use case is inserted into the behavior of the including use case. The first use case often depends on the outcome of the included use case. The notation is a dashed arrow from the base case to the included use case, with the label "«include»"

Use Case Diagram:



Activity Diagram:

An activity diagram visually displays a series of actions or the flow of control in a system. Using activity diagrams in business process modelling is common. Additionally, they can outline the procedures in a use case diagram. Activities might be sequential or concurrent while being modelled. An activity diagram will always have a start (the starting state) and a finish (the final state) in both scenarios.

There are several methods to represent activities, flows, decisions, guards, merging events, and more in between.

Initial State or Start Point:

Any activity diagram's first action state or starting point is shown as a tiny filled circle and an arrow.

Activity or Action State:

An action state is a representation of an object's continuously acting state.

Action Flow:

Action flows, also known as edges and routes, show how one action state changes into another. They are frequently represented with an arrowed line.

Synchronisation:

To divide a single incoming flow into numerous concurrent flows, a fork node is employed. In an activity diagram, it is shown as a straight, somewhat thicker line.

Swimlanes:

Swim lanes group related activities into one column.

Final State or End Point:

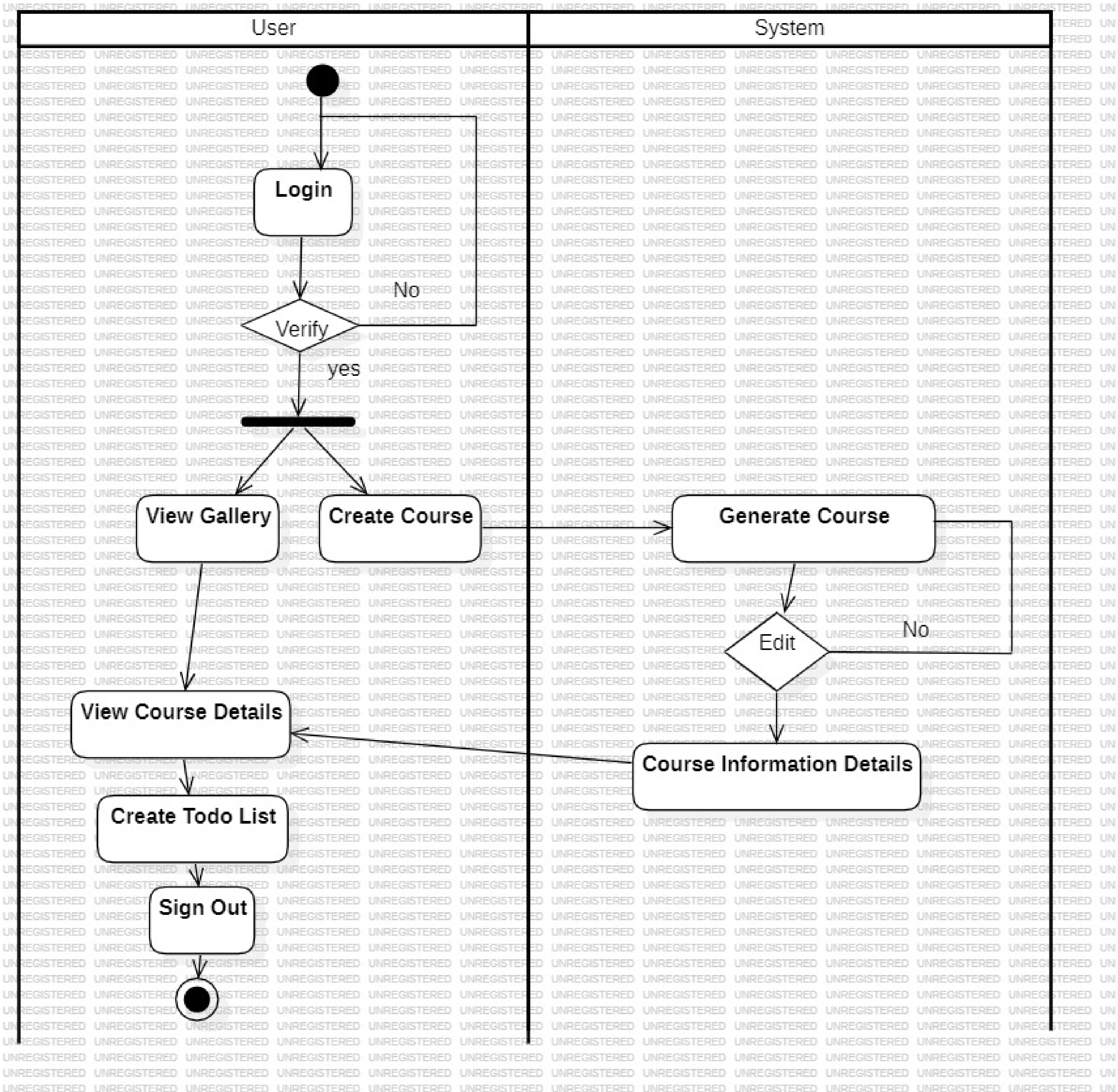
The ultimate action state is represented by an arrow pointing to a filled circle enclosed inside another circle.

Fork and Join Nodes:

Forks and joins have the same notation: either a horizontal or vertical bar (the orientation is dependent on whether the control flow is running left to right or top to bottom). They indicate the start and end of concurrent threads of control. The following diagram shows an example of their use.

A join is different from a merge in that the join synchronizes two inflows and produces a single outflow. The outflow from a join cannot execute until all inflows have been received. A merge passes any control flows straight through it. If two or more inflows are received by a merge symbol, the action pointed to by its outflow is executed two or more times.

Activity Diagram:



Database Diagram:

NoSQL databases come in a variety of types including document databases, key-values databases, wide-column stores, and graph databases. MongoDB is the world's most popular NoSQL database. The NoSQL database is a type of ER diagram that offers a more flexible design schema for data that doesn't fit into the rigid tabular relations used in other databases.

ELEMENTS OF ENTITY RELATIONSHIP DIAGRAM:

Entities:

Entities represent objects or concepts in the real world, which are stored in a database. Each entity has attributes that describe the properties or characteristics of the entity.

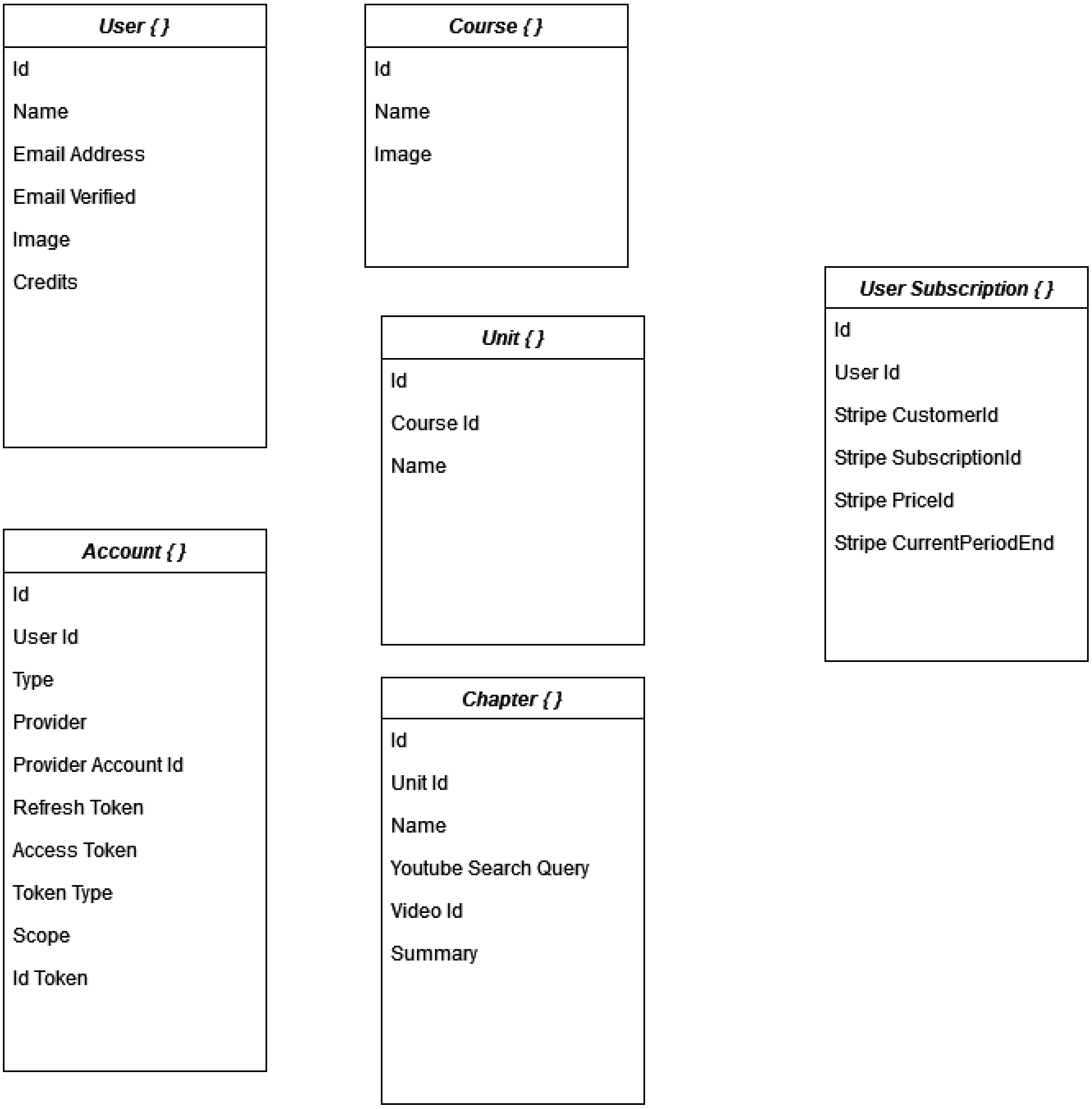
Attributes:

Attributes are properties or characteristics of an entity that hold data. They are typically represented as ovals or ellipses in the diagram and are connected to their respective entities. Attributes help define what kind of data is stored for each entity.

Relationships:

Relationships define how entities are connected or associated with each other in the database. Relationships show how data from one entity is related to data in another entity. They are typically
represented as lines connecting entities with diamond shapes that indicate the type of relationship.

Database Diagram:



User { }

Id
Name
Email Address
Email Verified
Image
Credits

Account { }

Id
User Id
Type
Provider
Provider Account Id
Refresh Token
Access Token
Token Type
Scope
Id Token

Course { }

Id
Name
Image

Unit { }

Id
Course Id
Name

Chapter { }

Id
Unit Id
Name
Youtube Search Query
Video Id
Summary

User Subscription { }

Id
User Id
Stripe CustomerId
Stripe SubscriptionId
Stripe PriceId
Stripe CurrentPeriodEnd

Class Diagram:

A class diagram is a type of static structure diagram in the Unified Modelling Language (UML) that represents the structure and relationships of classes in a system or software application. Class diagrams are commonly used in software engineering for modelling the classes, attributes, methods, and associations within a system's object-oriented design

ELEMENTS OF CLASS DIAGRAM:

Class

A class represents a blueprint or template for creating objects. It defines the attributes (data members or fields) and methods (functions or operations) that objects of the class will have.

Attributes

Attributes are properties or data members of a class that represent the characteristics or state of objects. They are typically shown as variables or fields inside the class rectangle.

Methods

Methods are functions or operations that can be performed on objects of the class. They are usually shown as functions inside the class rectangle.

Associations

Associations represent relationships or connections between classes.

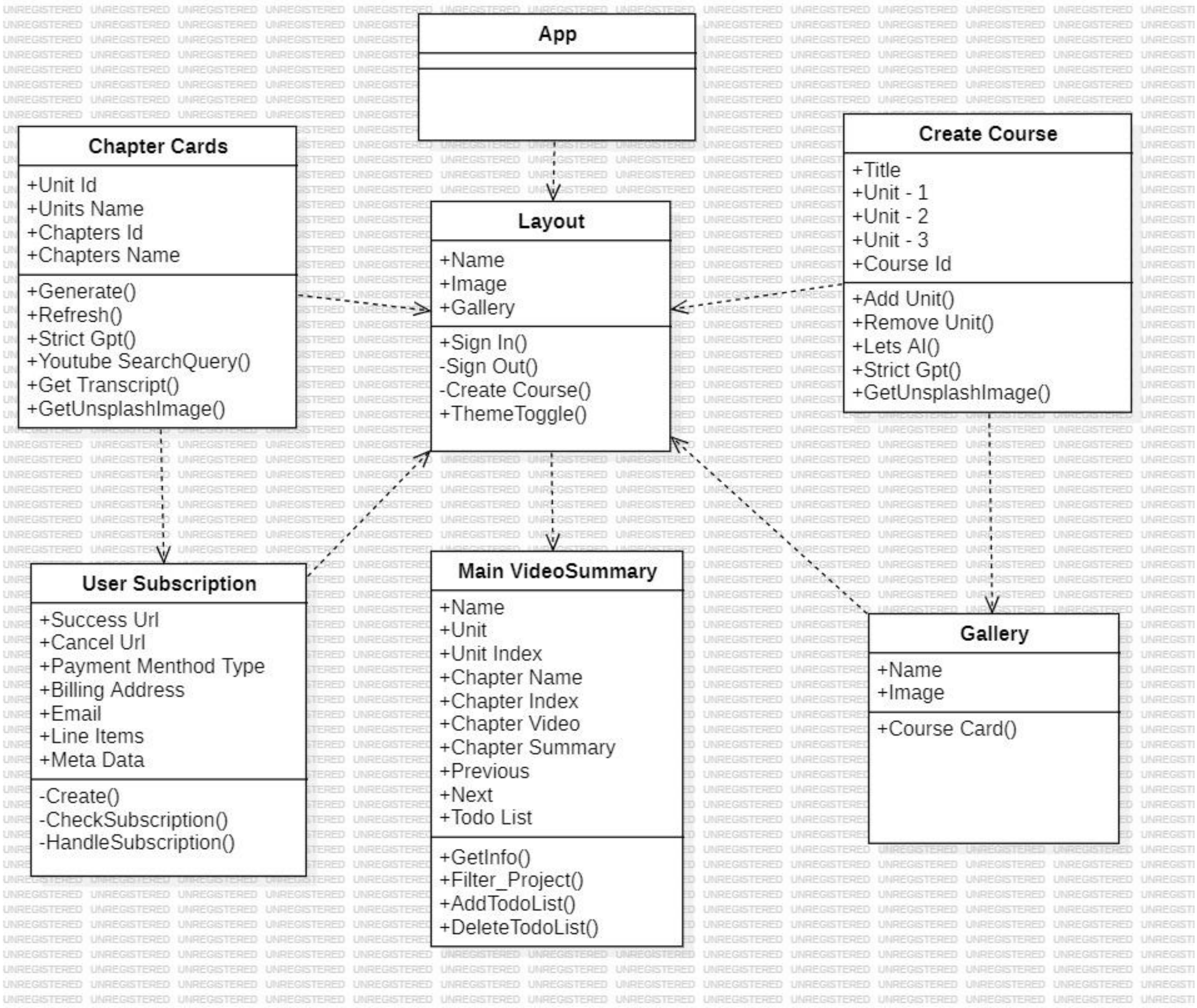
Inheritance (Generalization)

Inheritance represents the "is-a" relationship between classes. It shows that one class (the subclass or derived class) inherits the attributes and methods of another class (the superclass or base class).

Aggregation and Composition

Aggregation indicates a weaker relationship, where one class contains or is associated with other classes, but the contained objects can exist independently. Composition indicates a stronger relationship, where the whole (composite) class owns and manages its parts

Class Diagram:



SYSTEM IMPLEMENTATION

Test Case:

Test Case Id	Module	Form	Test condition	Steps of Test	Input Test Data	Expected output	Actual output	Status
T1	Login	User Login	To authenticate User using data from the google	1. Enter valid Email address 2. Tap on the login button	Email: email Password: password	Validated User and redirected to Website Landing Page	Validated User and redirected to Website Landing Page	Pass
T2	Login	User Login	To check the output when there is no Data entered.	1. Tap on the login button without entering email.	Email: (Empty) Password: (Empty)	Stay on Login Window and show “Please enter the email address” and “Please enter the password”.	Stay on Login Window and show “Please enter the email address” and “Please enter the password”	Pass
T3	Gallery	Gallery	To check the Course of multiple users.	1. Tap on the gallery button.	NA	Redirect to Course list page where you can see that Course details.	Redirect to Course list page where you can see that Course details.	Pass
T4	Create Course	Create Course	To add course of “AI”	1.Tap on the Create Course Button	Email: Krishna@gmail.com Password: Krishna@123	Redirect to Create Course Form page.	Redirect to Create Course Form page	Pass
T5	Create Course	Create Course Form	To add the Title of Course which user want to generate.	1.Tap on the input field Enter valid input.	Title: Frontend Development	Generate the details of relevant Units with 3 chapters related to the title of the course	Generate the details of relevant Units with 3 chapters related to the title of the course	Pass
T6	Create Course	Create Course Form	To add the Title and multiple Units of Course which user want to generate.	1.Tap on the input field Enter valid input.	Title: Frontend Development Unit 1: Introduction to Frontend Development Unit 2: JavaScript for Frontend Development Unit 3: Frontend Frameworks	Generate the details of relevant chapters related to the Title and Units of the course	Generate the details of relevant chapters related to the Title and Units of the course	Pass
T7	Chapters Cards	Chapters Cards box	To generate video and summary related to the chapters.	1.Tap on the continue button once all chapter’s cards turn green.	NA	Stay on Chapter Cards unit all chapter’s card box turn green	Stay on Chapter Cards unit all chapter’s card box turn green	Pass

T8	Main Video Summary	Main Video Summary	To see the video and summary of the specific units and chapters.	1.Tap on the video to play video.	NA	Show Video, summary related to chapters.	Show Video, summary related to chapters.	Pass
T9	Main Video Summary	Main Video Summary	To see the video and summary of the specific units and chapters.	1.Tap on the Next button.	NA	Move forward to next chapter	Move forward to next chapter	Pass
T10	Main Video Summary	Main Video Summary	To see the video and summary of the specific units and chapters.	1.Tap on the Previous button.	NA	Move backward to previous chapter	Move backward to previous chapter	Pass
T11	Todo Task	Todo Task	To see the pending tasks	1. To add the task, Enter the input field and press Enter button.	Task: HTML basic	Task has been added	Task has been added	Pass
T12	Todo Task	Todo Task	To Delete task which has been completed	1. To delete task, Enter clear button.	Task: HTML basic	Task has been deleted	Task has been deleted	Pass

Sequence Diagram:

sequence diagram is a type of Unified Modeling Language (UML) diagram that focuses on illustrating the interactions and chronological order of messages exchanged between objects or components within a system or between different systems. Sequence diagrams are commonly used in software engineering and system design to visualize the dynamic behavior of a system,particularly during the execution of a use case or scenario

ELEMENTS OF A SEQUENCE DIAGRAM:

Lifeline:

A vertical dashed line representing an object or participant involved in the interaction. Each lifeline corresponds to an instance of a class or component and spans the duration of its involvement in the sequence.

Activation Bar

An activation bar is a horizontal line that extends from a lifeline to represent the period during which an object is actively processing or executing a specific action or operation.

Message

Messages are arrows that show the flow of communication between objects or components. They indicate the exchange of information or the invocation of operations between lifelines.

Return Message

A return message (indicated by a dashed arrow with a label) shows the response from the receiver back to the sender after an operation is executed.

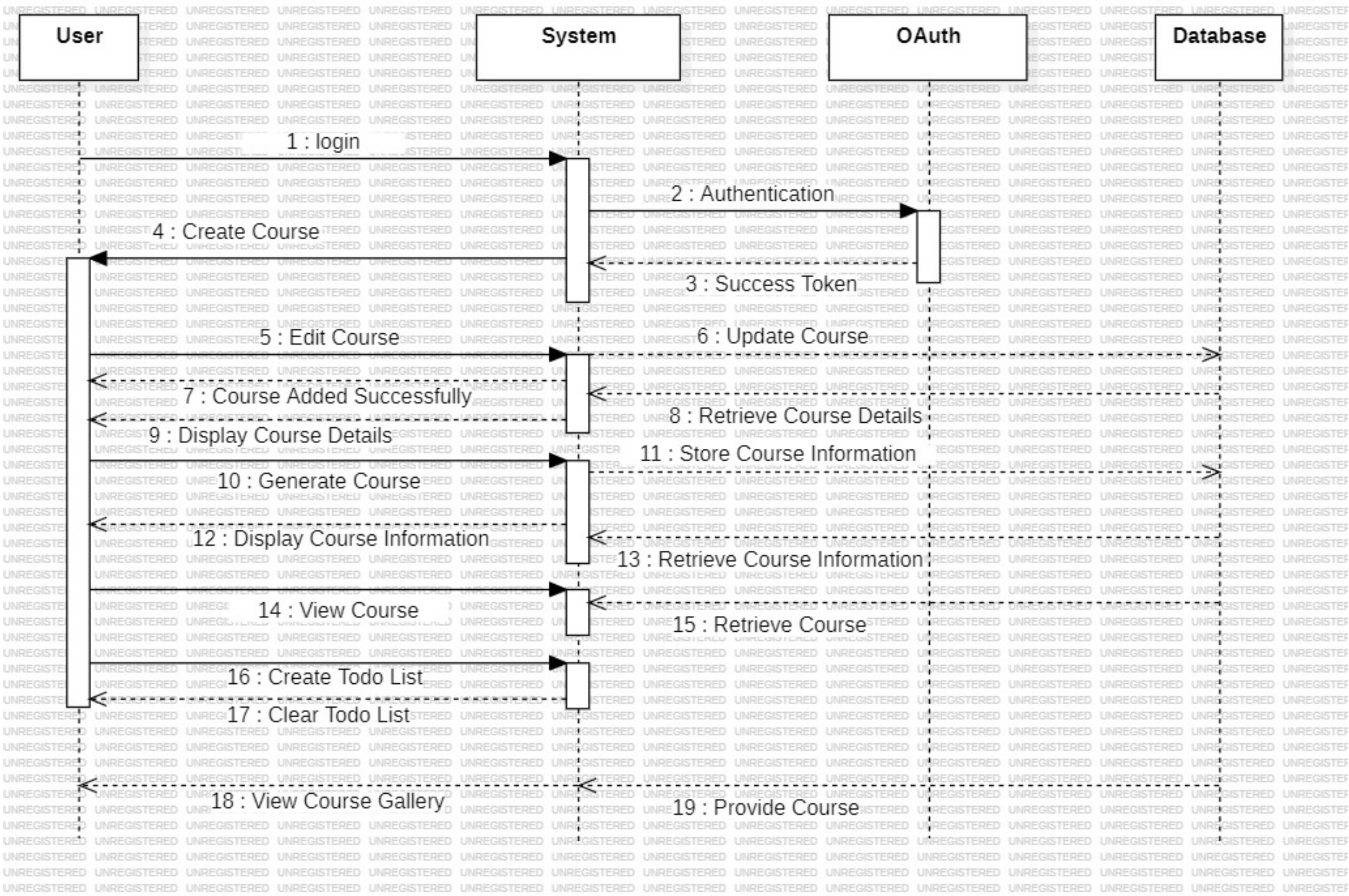
Synchronous Message

Denoted by a solid arrow, it represents a call where the sender waits for a response from the receiver before proceeding.

Asynchronous Message

Denoted by a dashed arrow, it represents a call where the sender doesn't wait for an immediate response and continues its execution

Sequence Diagram:



Component Diagram:

A component diagram is a type of Unified Modeling Language (UML) diagram used to represent the structural organization of a system or application into components, their relationships, and how they interact to form a complete system. Component diagrams are primarily used to model the high-level architecture of software systems and the dependencies between different parts of the system.

ELEMENTS OF A SEQUENCE DIAGRAM:

Components

Components are modular, reusable, and replaceable parts of a system. They can represent classes,packages, libraries, subsystems, or physical modules, depending on the context. Components are typically depicted as rectangles with the name of the component inside.

Interfaces

Interfaces represent the contract or set of services that a component provides or requires from other components. Interfaces are shown as circles on the boundaries of the component, and they can be labelled to indicate the specific services or methods exposed.

Dependencies

Dependencies between components are depicted as arrows pointing from one component to another. These arrows show that one component depends on another, which implies that changes in the depended-upon component can affect the dependent component.

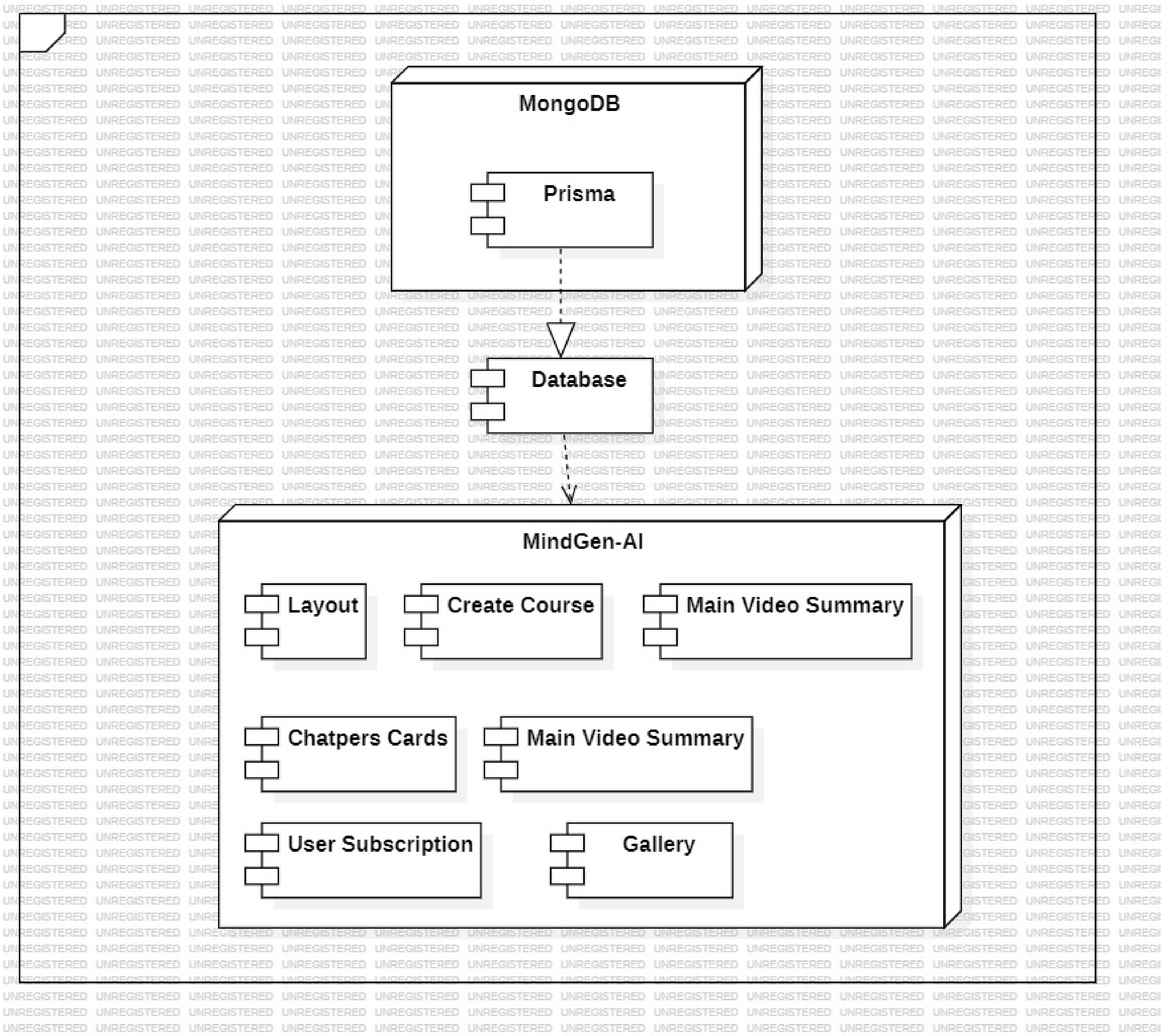
Associations

Associations between components can represent more flexible relationships between them, such as associations or connections between objects in object-oriented design. These are typically shown as lines connecting components without arrows.

Provided and Required Interfaces

Each component may have a set of provided interfaces (the services it offers) and required interfaces (the services it depends on from other components). Provided interfaces are usually shown inside the component, while required interfaces are shown as part of the component's boundary.

Component Diagram:



Deployment Diagram:

A deployment diagram is a type of Unified Modeling Language (UML) diagram that focuses on visualizing the physical deployment of software components and hardware nodes in a system or application. Deployment diagrams are used to illustrate how software artifacts (such as components, executables, and libraries) are distributed across hardware nodes (such as servers, computers, or devices) in a networked environment. These diagrams help software architects and developers understand how a system is deployed in the real-world infrastructure.

ELEMENTS OF A SEQUENCE DIAGRAM:

Nodes

Nodes represent hardware devices or execution environments in a deployment, such as servers, computers, routers, or physical devices. Nodes are typically depicted as rectangles with the name of the node inside.

Artifacts

Artifacts represent software components, executable files, or libraries that are deployed on nodes. They are shown as rectangles with the name of the artifact inside. Artifacts can be associated with nodes to indicate where they are deployed.

Communication Paths

Communication paths (also known as communication links or connectors) represent the connections and communication channels between nodes. Communication paths are represented by lines connecting nodes.

Associations

Associations between nodes and artifacts indicate that an artifact is deployed on a particular node.These associations are typically represented by a dashed line with an arrow pointing from the node to the artifact

.

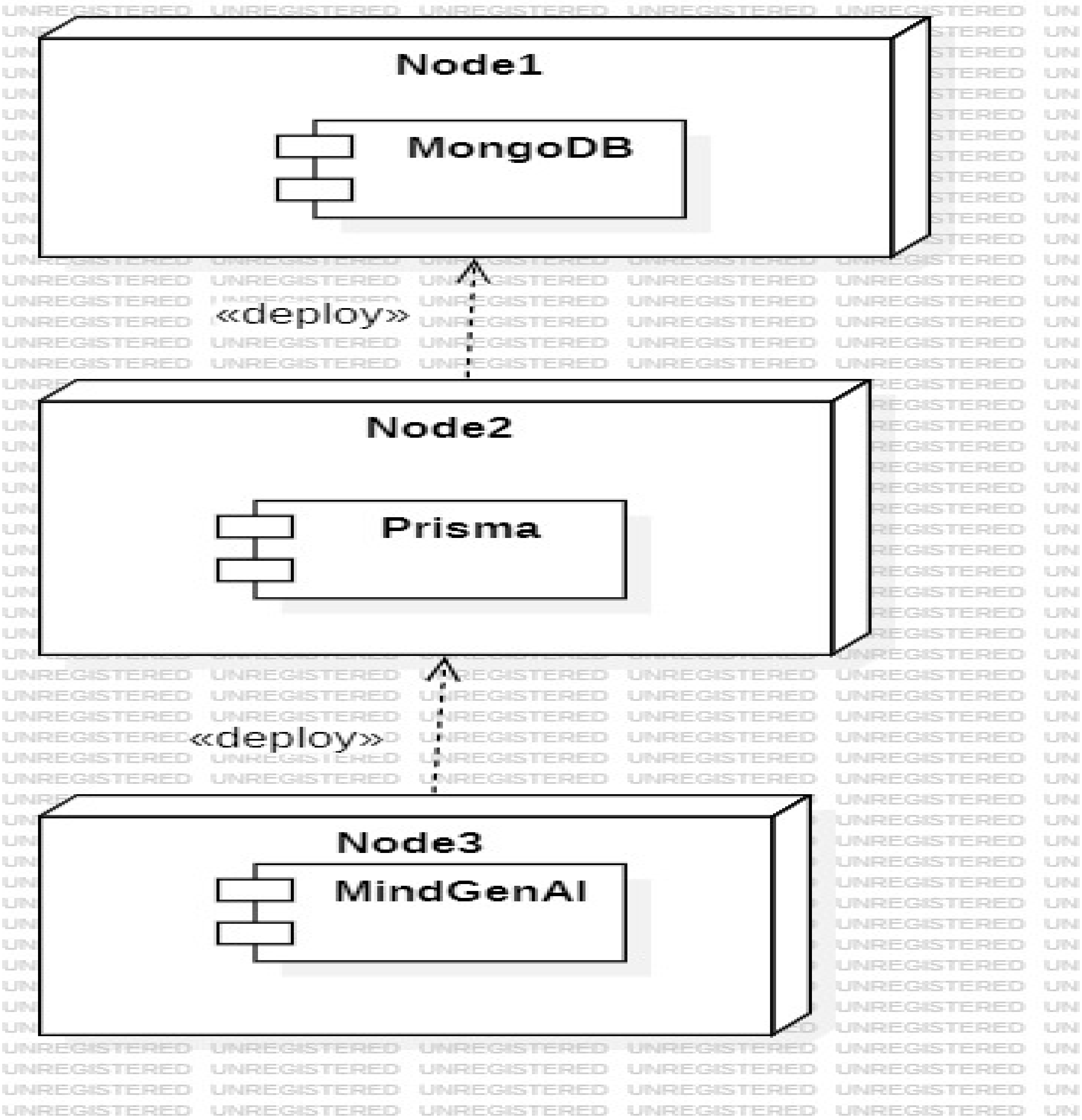
Deployment Relationships

Deployment relationships are used to specify how artifacts are deployed on nodes.
There are two main types of deployment relationships:

Deployment

Represents a one-to-one relationship between an artifact and a node, indicating that a specific artifact is deployed on a specific node

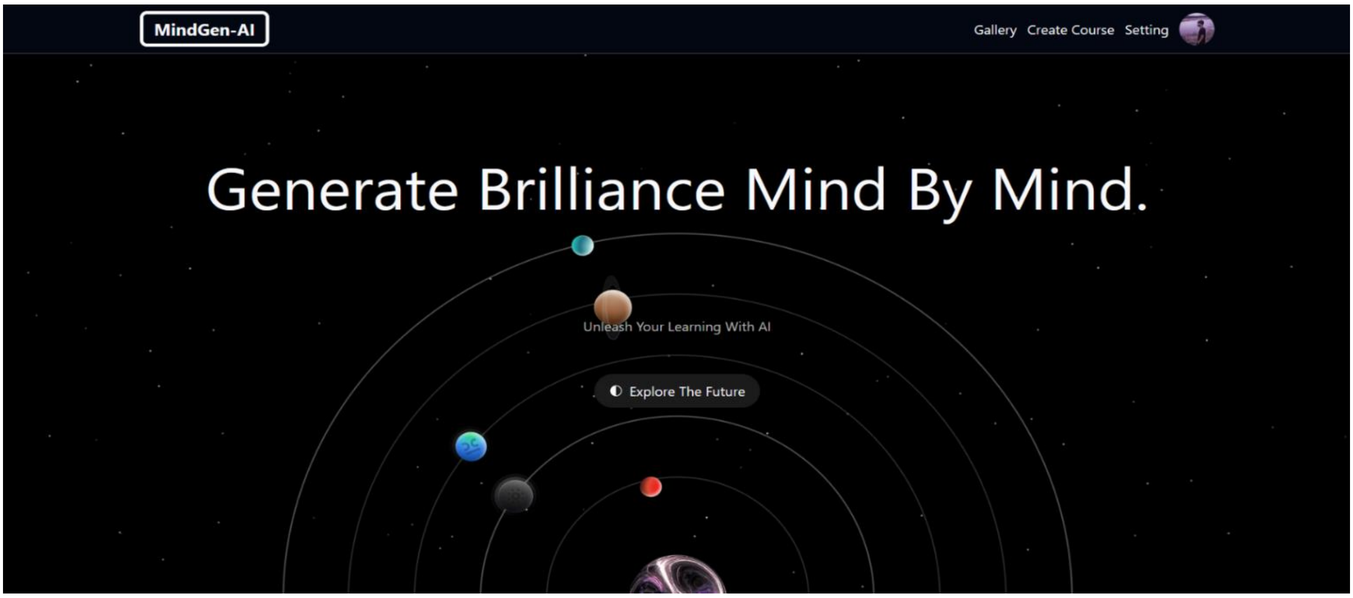
Deployment Diagram:



Images of Application:

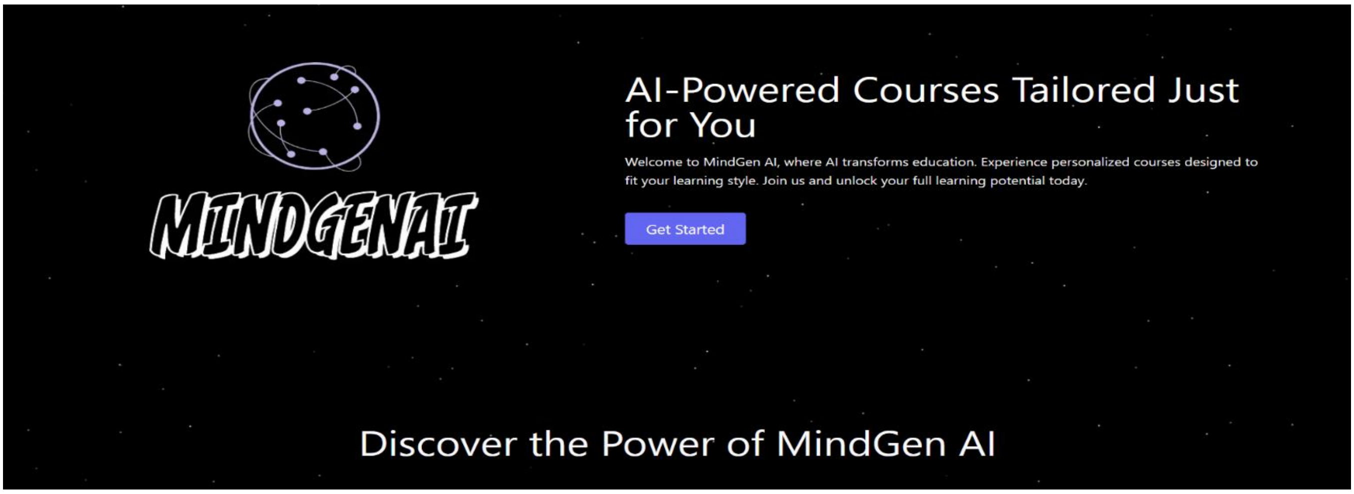
Landing Page:

Home Page:



Generate Brilliance Mind By Mind.

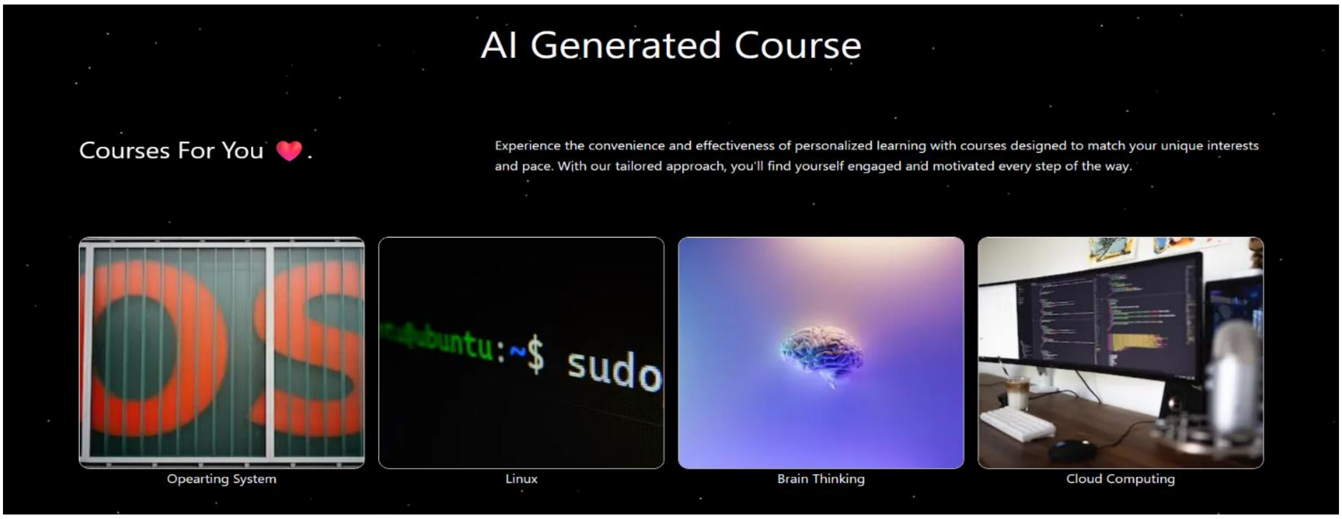
Unleash Your Learning With AI



AI-Powered Courses Tailored Just for You

Welcome to MindGen AI, where AI transforms education. Experience personalized courses designed to fit your learning style. Join us and unlock your full learning potential today.

Discover the Power of MindGen AI

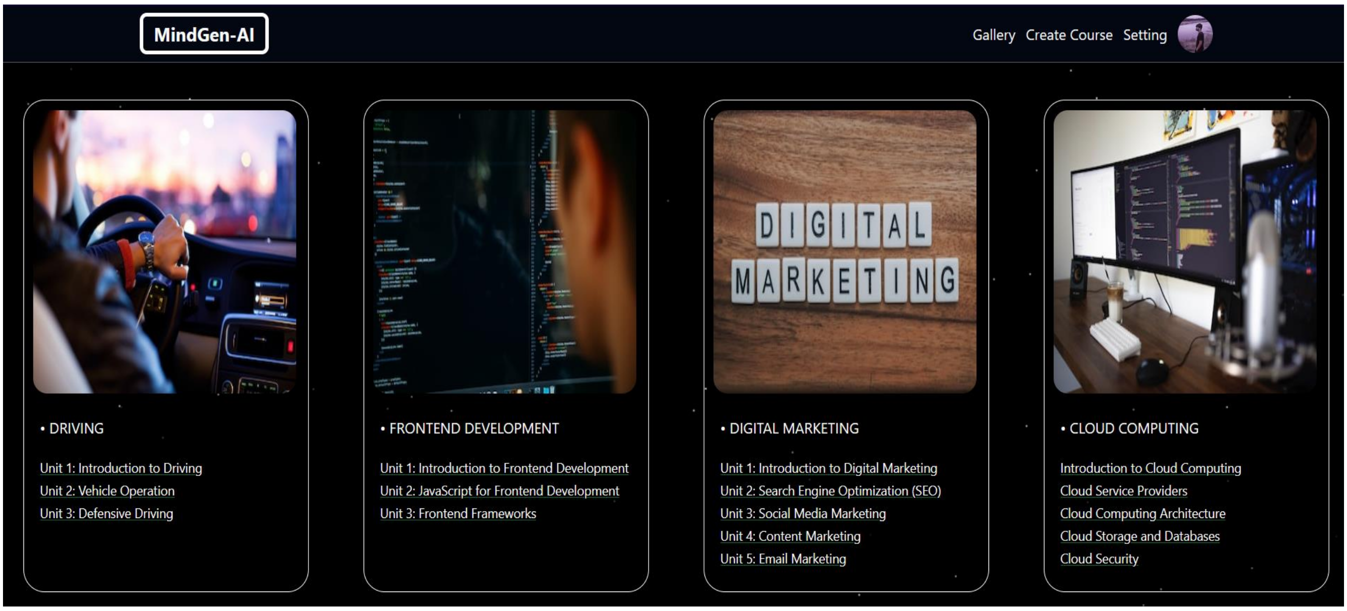


AI Generated Course

Courses For You ❤️ .

Experience the convenience and effectiveness of personalized learning with courses designed to match your unique interests and pace. With our tailored approach, you'll find yourself engaged and motivated every step of the way.

Gallery:



- DRIVING

Unit 1: Introduction to Driving
Unit 2: Vehicle Operation
Unit 3: Defensive Driving

- FRONTEND DEVELOPMENT

Unit 1: Introduction to Frontend Development
Unit 2: JavaScript for Frontend Development
Unit 3: Frontend Frameworks

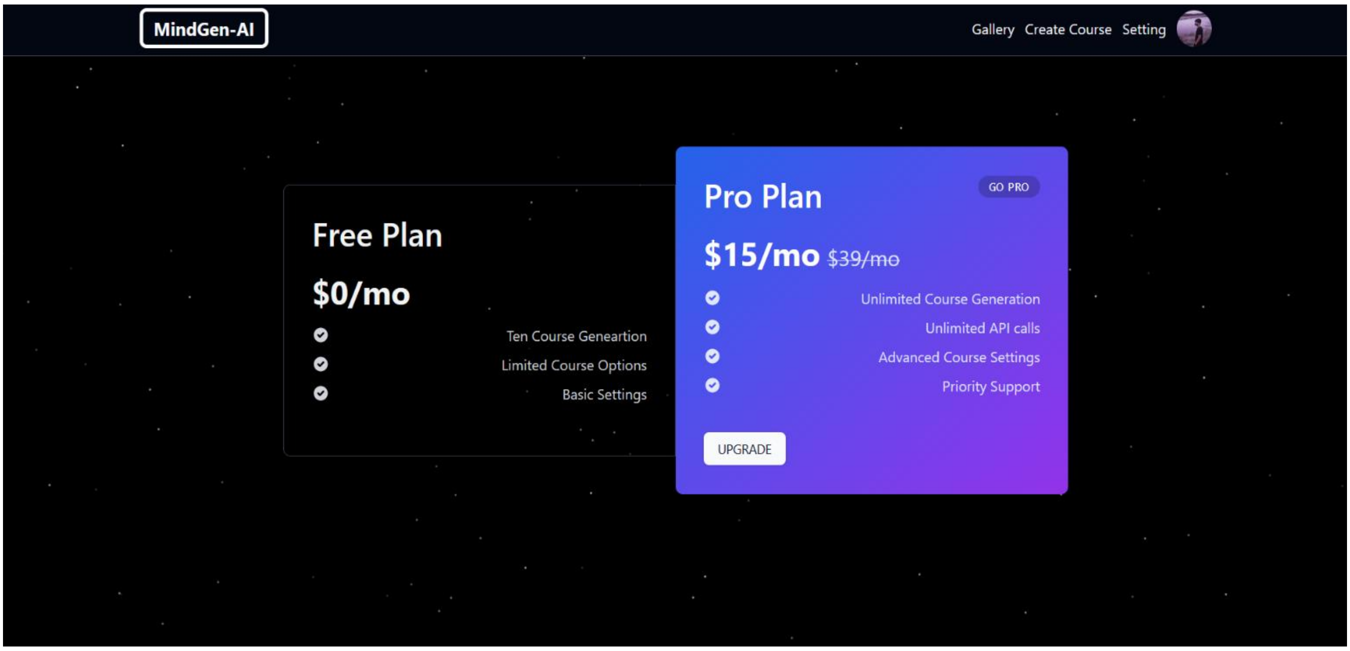
- DIGITAL MARKETING

Unit 1: Introduction to Digital Marketing
Unit 2: Search Engine Optimization (SEO)
Unit 3: Social Media Marketing
Unit 4: Content Marketing
Unit 5: Email Marketing

- CLOUD COMPUTING

Introduction to Cloud Computing
Cloud Service Providers
Cloud Computing Architecture
Cloud Storage and Databases
Cloud Security

Setting:



Free Plan

$0/mo

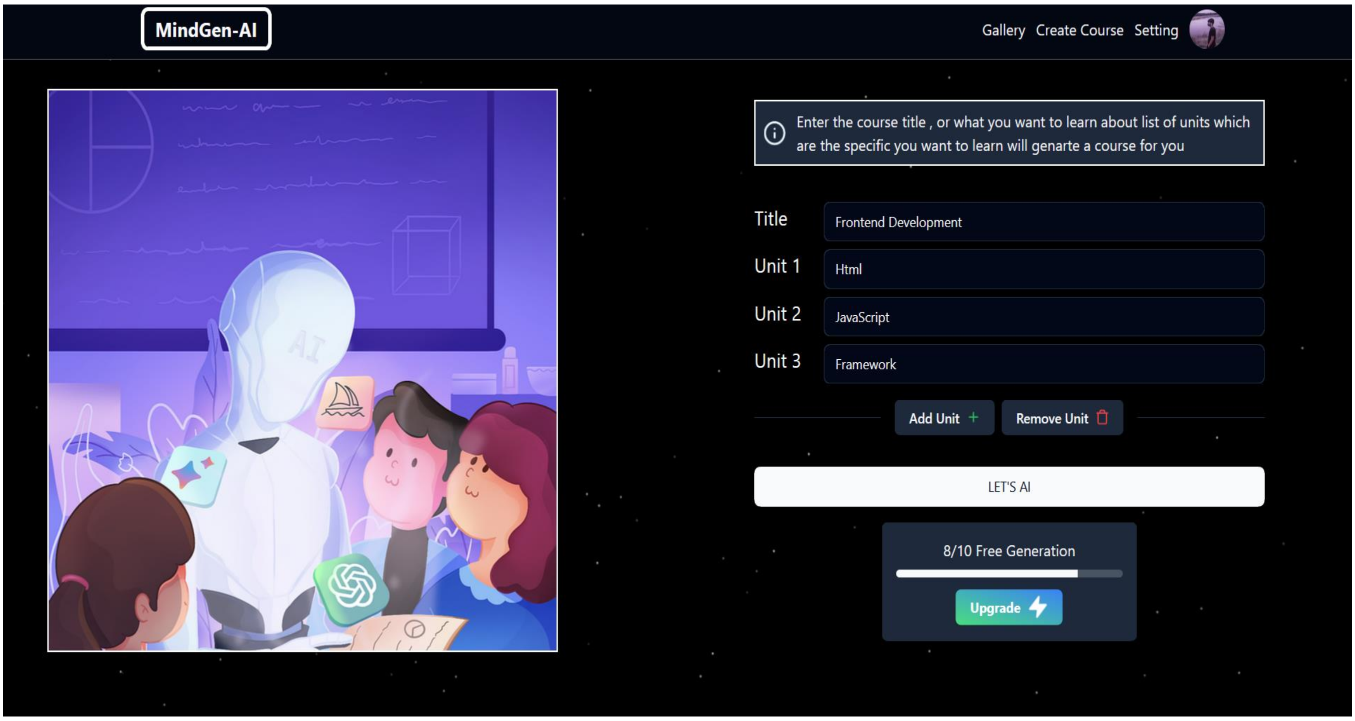
✓ Ten Course Geneartion
✓ Limited Course Options
✓ Basic Settings

Pro Plan

$15/mo ~~$39/mo~~

✓ Unlimited Course Generation
✓ Unlimited API calls
✓ Advanced Course Settings
✓ Priority Support

Create Course:



Enter the course title , or what you want to learn about list of units which are the specific you want to learn will genarte a course for you

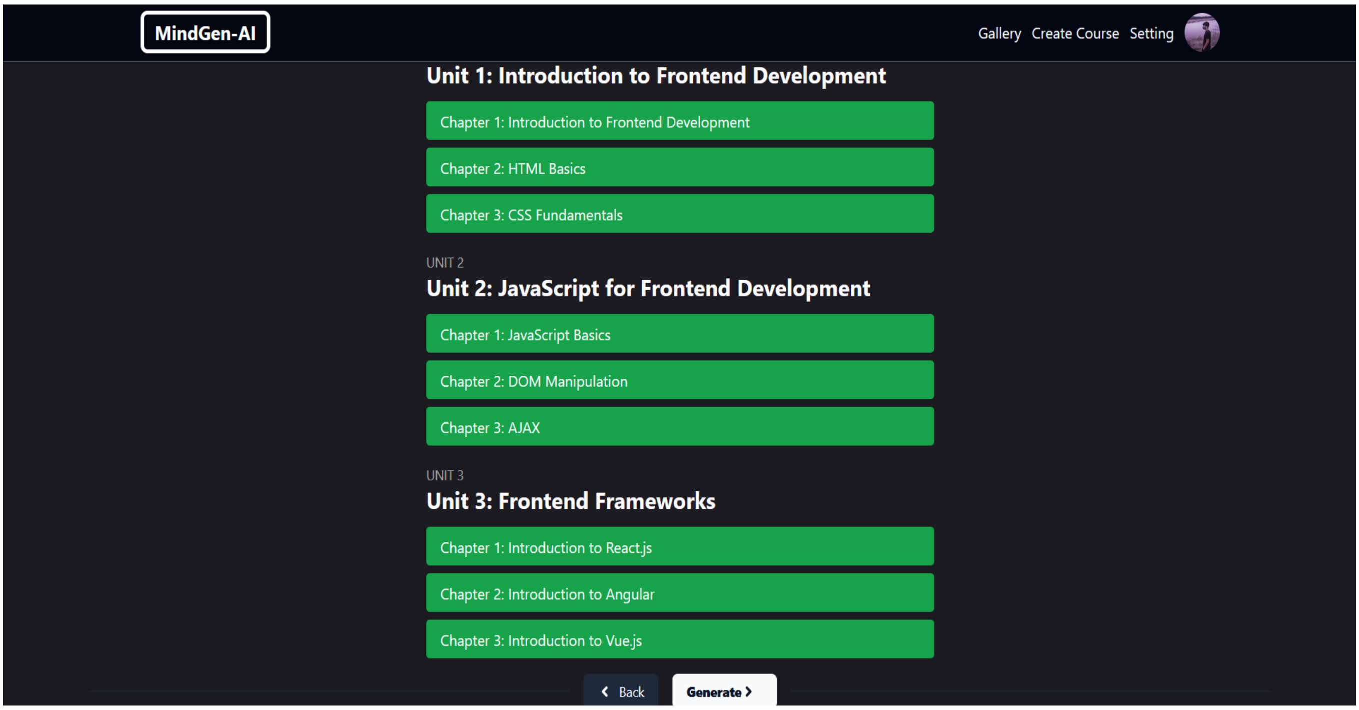
Title: Frontend Development

Unit 1: Html

Unit 2: JavaScript

Unit 3: Framework

Chapters Cards:



Unit 1: Introduction to Frontend Development

Chapter 1: Introduction to Frontend Development

Chapter 2: HTML Basics

Chapter 3: CSS Fundamentals

UNIT 2

Unit 2: JavaScript for Frontend Development

Chapter 1: JavaScript Basics

Chapter 2: DOM Manipulation

Chapter 3: AJAX

UNIT 3

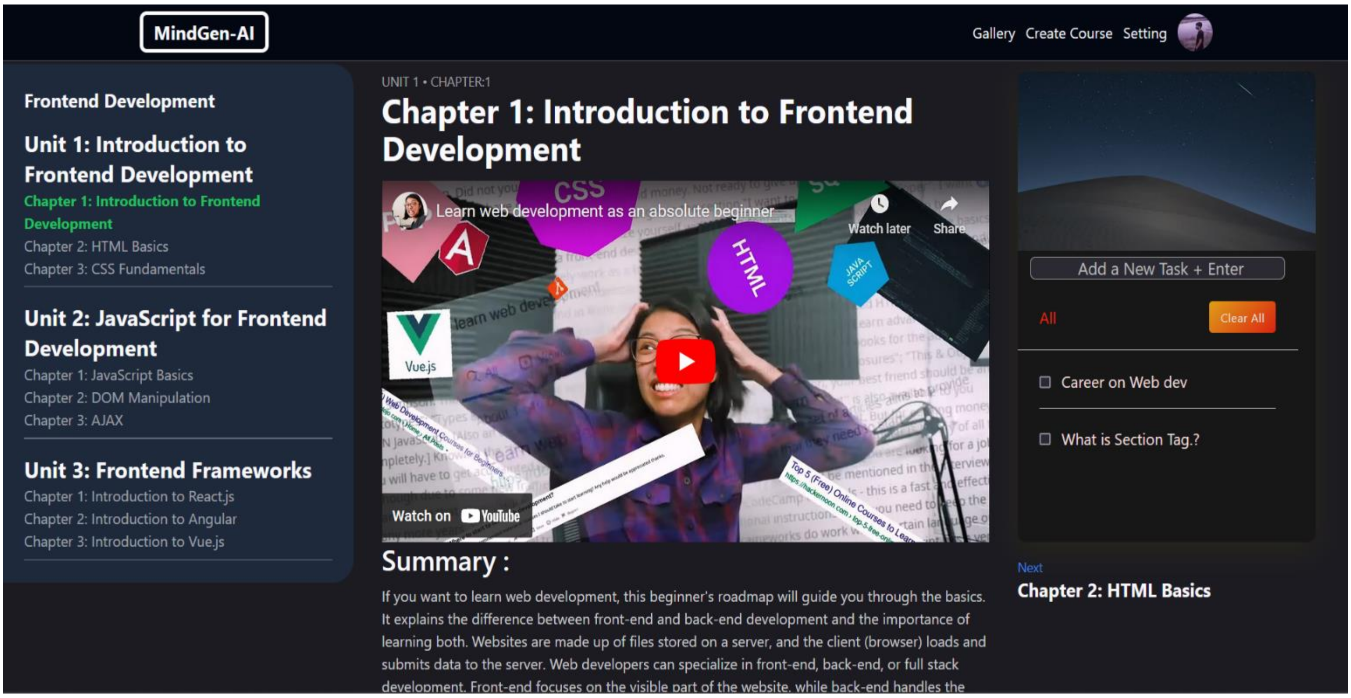
Unit 3: Frontend Frameworks

Chapter 1: Introduction to React.js

Chapter 2: Introduction to Angular

Chapter 3: Introduction to Vue.js

MainVideoSummary:



Frontend Development

Unit 1: Introduction to Frontend Development
Chapter 1: Introduction to Frontend Development
Chapter 2: HTML Basics
Chapter 3: CSS Fundamentals

Unit 2: JavaScript for Frontend Development
Chapter 1: JavaScript Basics
Chapter 2: DOM Manipulation
Chapter 3: AJAX

Unit 3: Frontend Frameworks
Chapter 1: Introduction to React.js
Chapter 2: Introduction to Angular
Chapter 3: Introduction to Vue.js

UNIT 1 • CHAPTER:1

Chapter 1: Introduction to Frontend Development

Summary :

If you want to learn web development, this beginner's roadmap will guide you through the basics. It explains the difference between front-end and back-end development and the importance of learning both. Websites are made up of files stored on a server, and the client (browser) loads and submits data to the server. Web developers can specialize in front-end, back-end, or full stack development. Front-end focuses on the visible part of the website, while back-end handles the

□ Career on Web dev

□ What is Section Tag.?

Chapter 2: HTML Basics

Future Enhancements:

MindGen AI is committed to continual improvement and growth. Future initiatives include expanding course offerings, enhancing AI capabilities, and exploring partnerships to broaden our impact on the global education landscape.

References:

Stack overflow:-

https://stackoverflow.com/

Design: -

https://diagram.com/

Prisma: -

https://www.youtube.com/watch?v=ReA5Jrlg

TypeScript: -

https://www.typescriptlang.org/

React.js: -
https://legacy.reactjs.org/docs/getting-started.html

CSS Design: -

https://codepen.io/