School of Information & Communication Technology
Gautam Buddha University
Greater Noida

Seminar report
Submitted in partial fulfillment of the requirements for the degree
Of
Bachelors of Technology
in
Computer Science & Engineering

By

Tanuja Tiwari
Roll No: 19/BCS/105
8th Semester (2022-23)

Under the Supervision of

Dr. Neeta Singh
Ms. Sumedha Dangi

Name of the Topic

Strapi

School of Information & Communication Technology
GAUTAM BUDDHA UNIVERSITY
GREATER NOIDA

CERTIFICATE

The Seminar Report entitled "**Strapi**" prepared and submitted by **Tanuja Tiwari (Roll Number: 19/BCS/105)**, has been found satisfactory in terms of scope, quality, and presentation as partial fulfillment of the requirements for the course CS- 490 (Seminar - 8th Semester) of the degree of **Bachelors of Technology in Computer Science and Engineering** in GAUTAM BUDDHA UNIVERSITY, GREATER NOIDA, INDIA.

(Dr. Neeta Singh)	**(Ms. Sumedha Dangi)**
Supervisor & Faculty of SoICT	Co-Supervisor & Faculty of SoICT

ACKNOWLEDGEMENT

The success and final outcome of learning Strapi required a lot of guidance and assistance from many people and I am extremely privileged to have got this all along the research of my topic and few of the projects which I made on the same. All that I have done is only due to such supervision and assistance and I would not forget to thank them. I respect and thank Ritik Sir, for providing me an opportunity to research on the topic and giving me all support and guidance. I am thankful to and fortunate enough to get constant encouragement, support and guidance from all everyone.

(Tanuja Tiwari)
19/BCS/105
B.Tech (CSE)
8th Semester
Gautam Buddha University

TABLE OF CONTENTS

INTRODUCTION

1.4 What is CMS?

- It stands for Content Management System
- A content management system (CMS) is software that helps users create, manage, and modify content on a website without the need for technical knowledge.
- CMS lets you build a website without needing to write code from scratch (or even know how to code at all).
- Examples of the most widely used open source CMS platforms include:
    a. WordPress
    b. Joomla
    c. Drupal
    d. Magento (e-commerce)
    e. PrestaShop (e-commerce

1.2 What is Strapi

- Strapi is a headless CMS that is used to develop websites, mobile applications, eCommerce sites, and APIs.
- It allows you to create an API without knowing anything about the backend or databases. The system builds APIs based on content models automatically, making it easy to view data in the CMS with Strapi examples.
- Strapi CMS is a free, open-source headless CMS that uses an API to link your frontend to Strapi's backend.
- It's a developer-friendly open-source and free-to-use service.
- Strapi is simple to learn and use, and you can get work done in minutes.
- Strapi may be used with React, Vue, Nuxt.js, Next.js, Angular, Svelte, Sapper, and Flutter.

1.3 Difference between Strapi and other CMS

- Strapi offers great flexibility and very granular (low-level) modeling of APIs, which may be created by this software.
- It has some drawbacks when it comes to migrating data and structure from classic CMSs (like WordPress). In most cases, a headless CMS requires a dedicated solution (so additional code development).
- Very subtle or significant modifications can be made easily, and most importantly, independently when working across multiple teams.
- The application is updated quite frequently, consisting of improvements, bug fixes, new features, which are great for overall security and futureproofing, but in rare cases may result in changes that break already developed APIs.
- Strapi doesn't support Typescript without custom modifications. It is planned to be introduced to the core, but there is no specific date for this feature yet.
- Developing and hosting production Strapi requires slightly more sophisticated DevOps resources compared to traditional CMS like WordPress running on PHP and Apache or Nginx.

1.4 Advantages of Using Strapi

1. Strapi is really simple to use, because of its user-friendly UI.
2. It's entirely configurable because of plugins.
3. Strapi is ready to use out of the box, but plugins allow you to customize it to your specific needs.
4. Strapi comes with an ORM that lets you create simple or complicated content types without having to write any code.
5. Strapi also provides an easy and versatile architecture for backend developers, with the ability to integrate third-party libraries like Koa and AngularJS 4.
6. Strapi will prove helpful if you wish to design a more sophisticated project. Its modular architecture allows you to personalize it and add your own features. As a result, the system is extremely adaptable to any project.
7. Strapi is easy to scale and can handle a high volume of traffic. Data validation is also quite speedy because of its interaction with Koa and Joi.
8. Strapi also features an automatic brown-field function that prevents developers from developing code that isn't being used. This allows them to concentrate on developing sophisticated features without having to worry about squandering computing resources.
9. You can not only use Strapi for free, but you can also contribute to it on GitHub.
10. As an open-source headless CMS project, it also has a strong developer community that works together to improve and expand the platform's capabilities.
11. This framework has a solid design that includes everything you'll need to create an API that your frontend application can easily use.
12. Strapi's modular architecture is responsible for all of its power. All of the plugins are installed via npm, which makes them extremely simple to set up. Each module also adheres to the Node.js standards and is designed to function with any server or framework.
13. It also has an admin part with a robust UI for managing content, which cuts development time in half.
14. Strapi offers a flexible authentication system that may be tailored to your specific requirements
15. Strapi not only offers a fantastic community, but it also has a fantastic collection of plugins that can be quickly put on your new or old app. Strapi also has certain unique plugins, such as Google Analytics integration, LDAP compatibility, and Swagger integrations, which set it apart from other CMS platforms.

1.5 Disadvantages of using Strapi

1. It's Difficult to Migrate Existing APIs - If you already have a backend system in place (for example, WordPress), moving it to Strapi will take some time if you want a perfect 100 percent match. Strapi, for example, does not have native support for WP-CRM. Therefore, you will have to manually deploy it in the background.
   That is why, rather than attempting to move something, most users utilize Strapi when starting from scratch.
2. TypeScript Support Is Limited - Strapi does not support TypeScript out of the box, which is a shame because it is a breakthrough technology. If you're an advanced user working on a large or complex project, this missing feature may hinder your experience.
3. Strapi isn't Completely Free and Open Source - Strapi is open source and free to use,

but it is not perfect. Strapi does, however, offer a Pro version with additional capabilities such as custom plugin support and professional assistance from their team with priority email access.

4. Continuous Updates - Strapi is a tool that is continually evolving, so expect frequent updates. Strapi's stable version receives weekly changes, so keep an eye on it if you don't want your platform to break unexpectedly. Although this is standard for most open-source tools these days, it is nonetheless inconvenient for developers who like stable software.

TECHNOLOGIES USED

2.1 Technologies/Terminologies associated with Strapi

- **Node JS**
    - Node.js is a cross-platform, open-source server environment that can run on Windows, Linux, Unix, macOS, and more.
    - Node.js is a back-end JavaScript runtime environment, runs on the V8 JavaScript Engine, and executes JavaScript code outside a web browser.
    - Node.js lets developers use JavaScript to write command line tools and for server-side scripting. The ability to run JavaScript code on the server is often used to generate dynamic web page content before the page is sent to the user's web browser.
    - Node.js allows the creation of Web servers and networking tools using JavaScript and a collection of "modules" that handle various core functionalities

- **Postgresql**
    - PostgreSQL also known as Postgres, is a free and open-source relational database management system (RDBMS) emphasizing extensibility and SQL compliance. It was originally named POSTGRES, referring to its origins as a successor to the Ingres database developed at the University of California, Berkeley.
    - PostgreSQL features transactions with Atomicity, Consistency, Isolation, Durability (ACID) properties, automatically updatable views, materialized views, triggers, foreign keys, and stored procedures.
    - It is designed to handle a range of workloads, from single machines to data warehouses or Web services with many concurrent users.

- **MySQL**
    - MySQL is an open-source relational database management system (RDBMS)
    - MySQL is written in C and C++.
    - It is a database management system.
    - The MySQL Database Server is very fast, reliable, scalable, and easy to use.

- **SQLite**
    - SQLite is a database engine written in the C programming language.
    - It is not a standalone app; rather, it is a library that software developers embed in their apps.
    - As such, it belongs to the family of embedded databases.
    - It is the most widely deployed database engine, as it is used by several of the top web browsers, operating systems, mobile phones, and other embedded systems.
    - SQLite was designed to allow the program to be operated without installing a

database management system or requiring a database administrator.

- **REST**
  - A REST API (also known as RESTful API) is an application programming interface (API or web API) that conforms to the constraints of REST architectural style and allows for interaction with RESTful web services.
  - REST stands for representational state transfer and was created by computer scientist Roy Fielding.
  - When a client request is made via a RESTful API, it transfers a representation of the state of the resource to the requester or endpoint. This information, or representation, is delivered in one of several formats via HTTP: JSON (Javascript Object Notation), HTML, XLT, Python, PHP, or plain text. JSON is the most generally popular file format to use because, despite its name, it's language-agnostic, as well as readable by both humans and machines.

- **GraphQL**

  - GraphQL provides a web API approach in which clients define the structure of the data to be returned by the server. This can impede web caching of query results.
  - GraphQL does not provide a full-fledged graph query language such as SPARQL, or even in dialects of SQL that support transitive closure. For example, a GraphQL interface that reports the parents of an individual cannot return, in a single query, the set of all their ancestors.
  - GraphQL consists of a type system, query language and execution semantics, static validation, and type introspection. It supports reading, writing (mutating), and subscribing to changes to data (realtime updates – most commonly implemented using Websockets). GraphQL servers are available for multiple languages. The result of a single query is returned in JSON format.

- **JWT**

  - Strapi uses JWT for authorization of resources.
  - JSON Web Token is a proposed Internet standard for creating data with optional signature and/or optional encryption whose payload holds JSON that asserts some number of claims. The tokens are signed either using a private secret or a public/private key.
  - JSON Web Tokens are an open, industry standard RFC 7519 method for representing claims securely between two parties.
  - A JWT consists of three parts: a header, payload, and signature.
    1. Header
       The header typically consists of two parts: the type of the token, which is JWT, and the algorithm that is used, such as HMAC SHA256 or RSA SHA256. It is Base64Url encoded to form the first part of the JWT.
    2. Payload
       The payload contains the claims. There is a set of registered claims, for example: iss (issuer), exp (expiration time), sub (subject), and aud (audience). These claims are not mandatory but recommended to provide a set of useful,

interoperable claims. The payload can also include extra attributes that define custom claims, such as employee role. Typically, the subject claim is used to create the OpenID Connect user subject. However, the Liberty JVM server can be configured to use an alternative claim. The payload is Base64Url encoded to form the second part of the JWT.

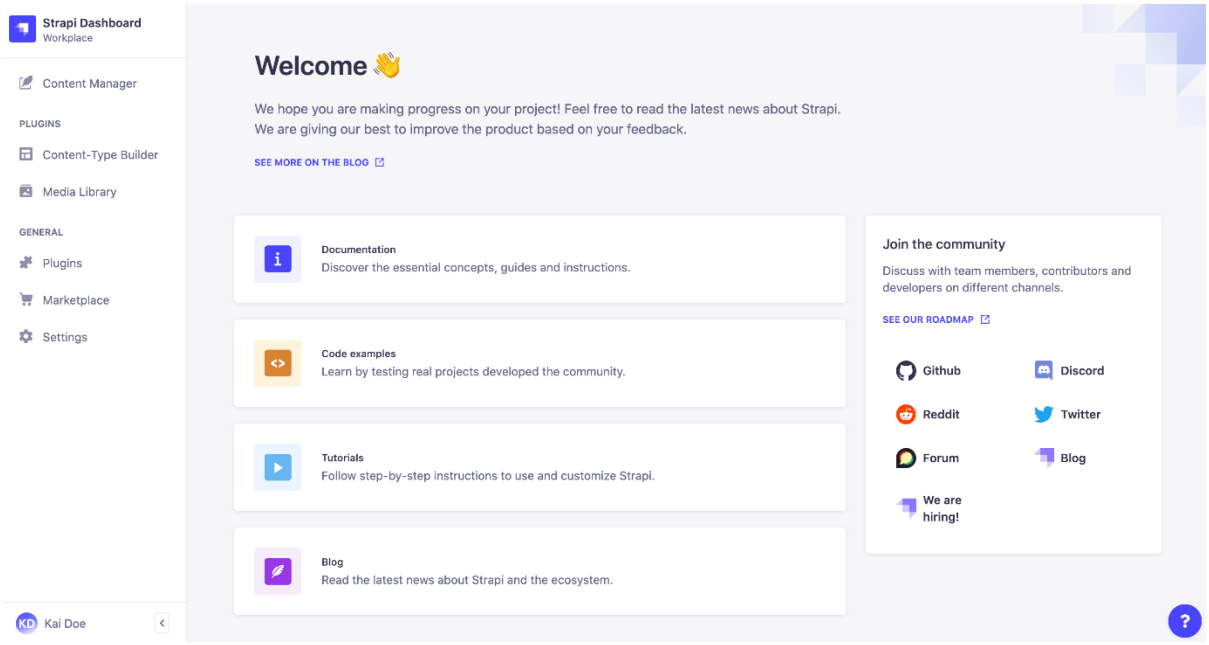
3. Signature
   To create the signature part, the encoded header and encoded payload are signed by using the signature algorithm from the header. The signature is used to verify that the issuer of the JWT is who it says it is and to ensure that the message wasn't changed along the way.

- **ORM**
  - Object Relational Mapping (ORM) is a technique used in creating a "bridge" between object-oriented programs and, in most cases, relational databases.
  - Put another way, you can see the ORM as the layer that connects object oriented programming (OOP) to relational databases.
  - When interacting with a database using OOP languages, you'll have to perform different operations like creating, reading, updating, and deleting (CRUD) data from a database. By design, you use SQL for performing these operations in relational databases.
  - While using SQL for this purpose isn't necessarily a bad idea, the ORM and ORM tools help simplify the interaction between relational databases and different OOP languages.

STRAPI DASHBOARD

3.1 Components of Strapi Dashboard

The Strapi dashboard looks like this-



- **Content Manager**

  The Content Manager is a core plugin of Strapi. It is a feature that is always activated by default and cannot be deactivated. It is accessible both when the application is in a development and production environment.

  The Content Manager is accessible from “Content Manager” in the main navigation, which opens a sub navigation displaying 2 categories: Collection types and Single types. Each category contains the available collection and single content-types which were created beforehand using the Content-type Builder. From these 2 categories, administrators can create, manage, and publish content.

- **Content Types Builder**

  The Content-type Builder is a core plugin of Strapi. It is a feature that is always activated by default and cannot be deactivated. The Content-type Builder is however only accessible when the application is in a development environment.

  Administrators can access the Content-type Builder from Content-type Builder in the main navigation of the admin panel.

  From the Content-type Builder, administrators can create and manage content-types: collection types and single types but also components.
  a. Collection types are content-types that can manage several entries.

b. Single types are content-types that can only manage one entry.
c. Components are a data structure that can be used in multiple collection types and single types.

All 3 are displayed as categories in the sub navigation of the Content-type Builder. In each category are listed all content-types and components that have already been created.

- **Media Library**

The Media Library is a Strapi plugin that is always activated by default and cannot be deactivated. It is accessible both when the application is in a development and production environment.

Administrators can access the Media Library from Media Library in the main navigation of the admin panel.

The Media Library displays all assets uploaded in the application, either via the Media Library itself or via the Content Manager when managing a media field. Assets uploaded to the Media Library can be inserted into content-types using the Content Manager.
From the Media Library, it is possible to:
a. upload a new asset (see adding assets) or create a new folder (see organizing assets with folders)
b. sort the assets and folders or set filters to find assets and folders more easily, toggle between the list view and the grid view to display assets, access settings to configure the view, and make a textual search
c. to find a specific asset or folder,
d. view, navigate through, and manage folders

- **User Roles and Permissions**

Some features of the admin panel, as well as the content managed with Strapi itself, are ruled by a system of permissions. These permissions can be assigned to roles, which are associated with the users who have access to the admin panel, the administrators. But it is also possible to grant permissions more publicly, to give access to content to the end users of your Strapi application.
Depending on what users and their roles and permissions you want to manage, you should either use the Role Based Access Control (RBAC) feature, or the Users & Permissions plugin. Both are managed from Settings, accessible from the main navigation of the admin panel.

- **Plugins**

Strapi is built around different types of plugins. Every default Strapi application comes with the following pre-installed plugins:
1. Content Manager

2. Content Type Builder
3. Email
4. Media Library (implemented via the Upload plugin)
5. Internationalization
6. Roles and Permissions

These plugins are essential for your Strapi application to function and cannot be uninstalled.

Additional plugins that you can use to extend and customize your Strapi applications are available in the Marketplace

From the admin panel, administrators are allowed to:

1. discover additional plugins and providers in the Marketplace (see Managing Marketplace plugins)
2. review the currently installed plugins and providers in Plugins

LEARNING OUTCOME

- Have a good understanding of the fundamental issues and challenges of Strapi.
- Have an understanding of the strengths and weaknesses of Strapi
- How to manage content using CMS.
- Learning all the tools associated with Strapi

REFERENCES

5.1 All Content used in this report is from

- https://www.wikipedia.org/
- https://www.edureka.co/
- https://www.javatpoint.com/
- https://medium.com/
- https://www.google.com/
- https://docs.strapi.io/

ANNEXURE

SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY
Gautam Buddha University, Greater Noida

Major Project, Internship & Seminar

Dated: 10/3/2023

This is to inform the Final Year students of B. Tech. (VIII Sem.) (CSE, AI, ECE and IT) that their **Major Project, Internship and Seminar Mid-Term Evaluation Presentation (offline mode) are scheduled on 6th April 2023, 10:00 am onwards**. All students please coordinate with their assigned Supervisors/mentors.

Program Name	Course Name	Code
B. Tech. CSE	Major Project, Seminar & Internship	CS492, CS490 & CS494
B. Tech. IT	Major Project, Seminar & Internship	IT492, IT490 & IT494
B. Tech. ECE	Major Project, Seminar & Internship	EC492, EC490 & EC494
B. Tech. AI	Major Project, Seminar & Internship	AI492, AI490 & AI494

- **All students should report strictly on time in the panel of their respective supervisors.**
- **All students must give project presentation (groupwise) and seminar (individual) and Internship presentation (individual) through power point.**
- **All students must submit spiral file for major project, seminar and Internship after the files are signed by their supervisors.**
- **All students must coordinate/contact with their supervisors for any other related queries.**

(Dr. Aarti Gautam Dinker)
(Coordinator, CSE/IT)

(Dr. Vimlesh Kumar Ray)
(Coordinator, ECE)

(Dr. Anurag Singh Baghel)
HoD, CSE

(Dr. Neeta Singh)
HoD, IT

Dr. Vidushi Sharma
HoD, ECE

(Prof. Sanjay Kumar Sharma)
Dean, USICT