

Grapes: NLP Web Craft

Project Proposal



Session: 2021-2025

Submitted by:

Naqeeb Ahmed

2021-CS-616

Munib Ur Rehman

2021-CS-605

Ahsan Bashir

2021-CS-628

Supervised by:

Dr. Zeeshan Ramzan

Department of Computer Science, New Campus
University of Engineering and Technology Lahore, Pakistan

Contents

1	Abstract	3
2	Introduction	3
3	Problem Statement	5
4	Objectives	6
5	Scope and Features	7
5.1	Key Features	7
6	Tools and Technologies	8
6.1	Software Tools and Technologies	8
6.2	Hardware Tools and Technologies	8
7	Proposed Methodology/System	9
7.1	Steps	9
7.1.1	Requirements Gathering	9
7.1.2	Design	9
7.1.3	Steps Planning	9
7.1.4	Development	9
7.1.5	Testing	10
7.1.6	Iteration and Incrementation	10
8	Related Work	11

List of Figures

1	Methodology	11
2	Front end of the Jimdo system	12
3	Front end of the CodeWP system	13
4	Front end of the Zyro system	14
5	Front end of the Grid system	15

6	Front end of the Webflow system	16
7	Gantt Chart	20

List of Tables

1	Comparison of Website Builders and AI Integration (Part 1)	17
2	Comparison of Website Builders and AI Integration (Part 2)	18
3	Team Members and Individual Tasks	19

1 Abstract

In today's digital era, the demand for e-commerce, blogs, and portfolio websites is growing rapidly, yet the process of creating them remains complex and time-consuming for many individuals and businesses. The proposed AI Web Builder solution aims to simplify this challenge by providing an intuitive platform powered by AI algorithms. The primary objectives of the project include streamlining the website creation process, reducing the need for manual coding, and enabling users with limited technical knowledge to design professional and responsive e-commerce stores, blogs, and portfolio websites. The platform will feature a user-friendly interface with AI capabilities, allowing users to customize layouts, choose design elements, and optimize content for their specific needs. Implementation will follow an iterative design and development approach, incorporating user feedback to refine the AI algorithms and improve overall usability. The platform will be evaluated through user testing, focusing on ease of use, design satisfaction, and website performance. This project has the potential to democratize web design for e-commerce, blogs, and portfolios, making it more accessible and stimulating innovation in the digital space.

2 Introduction

In today's digital age, the ability to create captivating and functional websites is essential for individuals and businesses, particularly for e-commerce, blogs, and portfolio sites. However, the complexity of traditional web development processes often acts as a barrier for those without technical expertise. To address this challenge, our final year project proposes the development of an innovative AI web builder tailored for these specific website types. This project aims to democratize website creation by leveraging cutting-edge technologies such as chatbots, Natural Language Processing (NLP) models, GrapesJS, and Python backend integration. By combining these technologies, we seek to provide users with an intuitive and intelligent platform that simplifies the creation of e-commerce stores, blogs, and portfolio websites, regardless of their technical background.

The AI web builder project starts with a clear vision: to empower users to create websites effortlessly. At its core lies a sophisticated chatbot powered by TensorFlow and Keras Sequential Model, seamlessly integrated into a Flask or Django backend. This

chatbot serves as the user’s personal assistant, guiding them through the website creation journey with intelligence and ease. Users interact with the chatbot by entering prompts, and the system extracts keywords from these prompts to understand user intent and provide relevant guidance.

One of the key advantages of integrating a chatbot into the AI web builder is its accessibility. Chatbots provide an intuitive interface that is accessible to users of all skill levels. Whether they are seasoned developers or complete beginners, users can navigate the website creation process with ease, thanks to the assistance provided by the chatbot. Additionally, chatbots offer efficiency by streamlining the website creation process. With AI-powered assistance, users can quickly and efficiently build their websites, saving time and effort in the process.

In addition to the chatbot, our AI web builder project leverages the power of NLP models to further enhance the user experience. NLP models play a crucial role in understanding user input and extracting key information from prompts. By analyzing the text provided by users, NLP models can identify user intent, extract relevant keywords, and provide contextually relevant responses. This enables more natural and effective communication between the user and the AI web builder, enhancing the overall user experience.

The next phase involves the integration of GrapesJS into the AI web builder project, offering another layer of functionality and flexibility. GrapesJS is a powerful tool that allows users to visually edit website layouts without the need for coding knowledge. With GrapesJS, users can easily add, remove, and rearrange elements on their website, experimenting with different layouts and designs until they find the perfect fit. This visual editing capability empowers users to take control of their website’s design, making the website creation process more intuitive and enjoyable.

Furthermore, GrapesJS facilitates real-time updates, allowing users to see the immediate impact of their edits as they make them. This instant feedback loop enables users to iterate quickly and refine their website designs until they are satisfied with the results. GrapesJS also offers extensive customization options, allowing users to personalize every aspect of their website to suit their unique preferences and requirements. From selecting responsive templates to managing website content, GrapesJS provides users with the tools they need to create stunning and functional websites effortlessly.

Python backend integration is another essential component of our AI web builder project. Python’s versatility and robustness make it an ideal choice for implementing backend functionality such as database management, user authentication, and content management. By integrating Python into the AI web builder, we ensure that the system can handle dynamic website functionality effectively. This includes managing user accounts, storing website data, and providing seamless integration with third-party services and APIs.

Moreover, Python backend integration enables scalability, allowing the AI web builder to handle increasing user demands and complexities over time. As the user base grows and the system becomes more sophisticated, Python backend ensures that the AI web builder can scale to meet these evolving needs. Additionally, Python’s extensive library ecosystem provides access to a wide range of tools and resources that can further enhance the functionality and performance of the AI web builder.

The integration of chatbots, NLP models, GrapesJS, and Python backend creates a comprehensive and intelligent platform that empowers users to create stunning websites effortlessly. This platform offers accessibility, efficiency, personalization, and continuous improvement, ensuring that users can build websites that meet their unique needs and preferences.

In summary, the integration of advanced technologies into our AI web builder project provides numerous benefits to users. From intuitive chatbot guidance to the flexibility of GrapesJS for visual design and Python’s robust backend functionality, this project aims to simplify and revolutionize the website creation process. By democratizing access to web design tools, this AI web builder has the potential to drive innovation and broaden accessibility in the digital space.

3 Problem Statement

To develop an AI-based system using artificial intelligence, GrapesJS, and NLP technology to create attractive, functional websites without the need for extensive coding or design expertise.

4 Objectives

The end goal is to provide users with a user-friendly platform where they can easily design and customize websites tailored to their specific needs and preferences.

- Increase user satisfaction and ease of use by making the website creation process more accessible for users with varying technical abilities.
 - Conduct user surveys or usability tests to assess user satisfaction and perceived ease of use before and after implementing the AI web builder.
 - Measure changes in user satisfaction scores or completion rates of website creation tasks by users with varying technical abilities.
- Reduce reliance on coding and technical knowledge for website creation, allowing users to focus on design and content creation.
 - Track the percentage of users who successfully create websites using the AI web builder without needing to write custom code or seek external technical assistance.
 - Compare this percentage to baseline data before implementing the AI web builder.
- Increase user engagement and creativity in web design by providing access to advanced features, templates, and designs.
 - Monitor user engagement metrics such as the number of templates and designs explored, features utilized, and customization options exercised by users.
 - Analyze the frequency and depth of user interactions with advanced features to gauge their impact on creativity and experimentation.
- Ensure that websites created with the AI web builder are responsive and accessible across different devices and screen sizes.
 - Conduct cross-device compatibility tests to assess the responsiveness and accessibility of websites created using the AI web builder.

- Measure the percentage of websites that pass compatibility tests and provide an optimal user experience across various devices and screen sizes.
- Enhance user interaction and understanding through Natural Language Processing (NLP), allowing users to communicate with the AI web builder in natural language.
 - Evaluate the accuracy of NLP understanding by tracking the percentage of user prompts correctly interpreted by the AI web builder.
 - Measure the improvement in accuracy over time through iterations and updates to the NLP model.

5 Scope and Features

The scope of the project encompasses the design, development, and implementation of the AI web builder platform, including:

5.1 Key Features

- **Drag-and-Drop Interface:** An intuitive interface allowing users to easily add, remove, and rearrange elements on their website without coding.
- **AI-Powered Design Suggestions:** AI algorithms suggesting design elements, layouts, and color schemes based on user preferences and trends.
- **Responsive Templates:** A selection of responsive templates ensuring websites look great on various devices and screen sizes.
- **Content Management System (CMS):** A built-in CMS for managing website content, including text, images, and multimedia.
- **Intent Recognition:** Implementing NLP models to recognize the intent behind user prompts, allowing the system to understand user requirements and preferences more accurately.
- **Keyword Extraction:** Utilizing NLP techniques to extract keywords from user prompts, enabling the system to identify key elements and topics relevant to website creation.

- **Voice Input Support:** Enabling users to interact with the AI web builder using voice commands, allowing them to dictate prompts and commands instead of typing.

6 Tools and Technologies

6.1 Software Tools and Technologies

The following software tools and technologies will be utilized for the project:

Programming Languages:

- HTML
- CSS
- Bootstrap
- Flask or Django
- Python
- GrapesJS

Software Tools:

- Pencil
- Visual Studio Code
- StarUML
- MS Office
- MS Visio

6.2 Hardware Tools and Technologies

- 64-bit Windows 11 Operating System
- 8GB RAM
- Intel(R) Core(TM) i5-6300U CPU @ 2.40 GHz

7 Proposed Methodology/System

We are employing an agile methodology for the project. The development of the AI Web builder will follow an iterative and incremental approach, focusing on user feedback and continuous testing. Both web and machine learning (NLP) technologies will be utilized throughout the process.

7.1 Steps

7.1.1 Requirements Gathering

Engage stakeholders to identify project goals, user needs, and requirements for both the web app and NLP integration. Create a backlog of user stories and features, prioritized based on business value and urgency.

7.1.2 Design

Collaborate with the development team to create wireframes, mockups, and design prototypes for the web app's frontend and backend. Design the architecture and components required for NLP integration, including data preprocessing, model training, and inference.

7.1.3 Steps Planning

Select high-priority items from the backlog for the upcoming sprints, considering both web app and NLP tasks. Break down selected items into smaller tasks and estimate the effort required for each task. Set sprint goals and duration, ensuring a balance between web app development and NLP integration.

7.1.4 Development

- Implement the frontend of the web app using HTML, CSS, and JavaScript, focusing on creating user-friendly interfaces for website creation and NLP interaction.
- Develop backend functionality using Flask or Django, including database management, user authentication, and integration with the NLP model.
- Train and fine-tune the NLP model using a dataset optimized for website creation prompts.

- Integrate the trained NLP model into the backend of the web app, enabling natural language interaction for users.

7.1.5 Testing

- Conduct unit testing for individual components of the web app and NLP integration to ensure they function correctly and meet specified requirements.
- Perform integration testing to validate the interaction between the web app frontend, backend, and NLP model.
- Implement automated testing for regression testing and continuous integration to catch bugs early in the development process.
- Conduct user acceptance testing (UAT) to gather feedback from stakeholders and end-users, refining the web app and NLP functionality based on their input.

7.1.6 Iteration and Incrementation

After each sprint, review the progress made and adjust the backlog based on stakeholder feedback and changing requirements. Plan the next sprint, incorporating new features, enhancements, and bug fixes based on lessons learned from previous iterations. Continuously iterate and increment on both the web app and NLP integration, delivering value to users with each cycle.

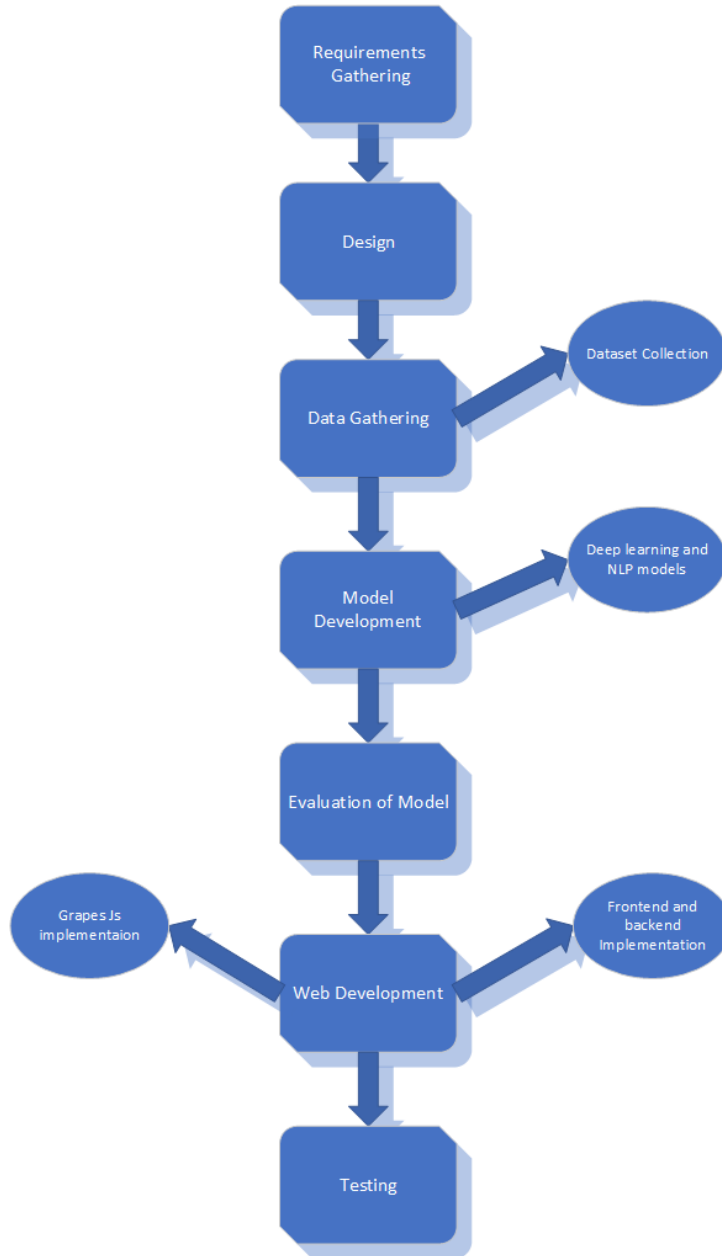


Figure 1: Methodology

8 Related Work

Wix [1]: Wix is a popular website builder that offers a variety of features, including an AI-powered website builder called Wix ADI (Artificial Design Intelligence). Wix ADI asks you a few questions about your business or website goals, and then it automatically generates a website for you. You can then customize the website to your liking using the Wix editor.

Jimdo [2]: Jimdo offers a free AI website builder that is easy to use and perfect for beginners. Jimdo's AI builder asks you a few questions about your website, and then it creates a website for you with pre-written content and images. You can then customize the website to your liking using the Jimdo editor. The front end of the system Jimdo is shown in Figure 2.

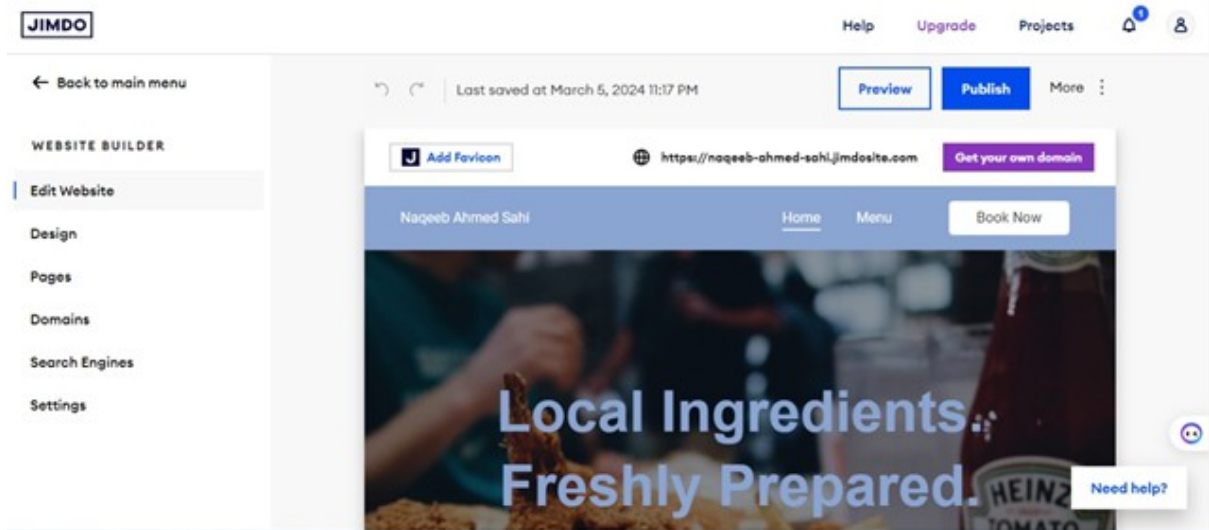


Figure 2: Front end of the Jimdo system

Unbounce [3]: Unbounce is an AI-powered landing page builder that helps you create high-converting landing pages for your marketing campaigns. Unbounce's AI features include Smart Copy, which helps you write persuasive copy for your landing pages, and Smart Traffic, which helps you send your landing page traffic to the right audience.

CodeWP [4]: CodeWP is an AI-powered tool that helps you build complex WordPress websites. CodeWP's AI features include content generation, code generation, and SEO optimization. The front end of the system CodeWP is shown in Figure 3.

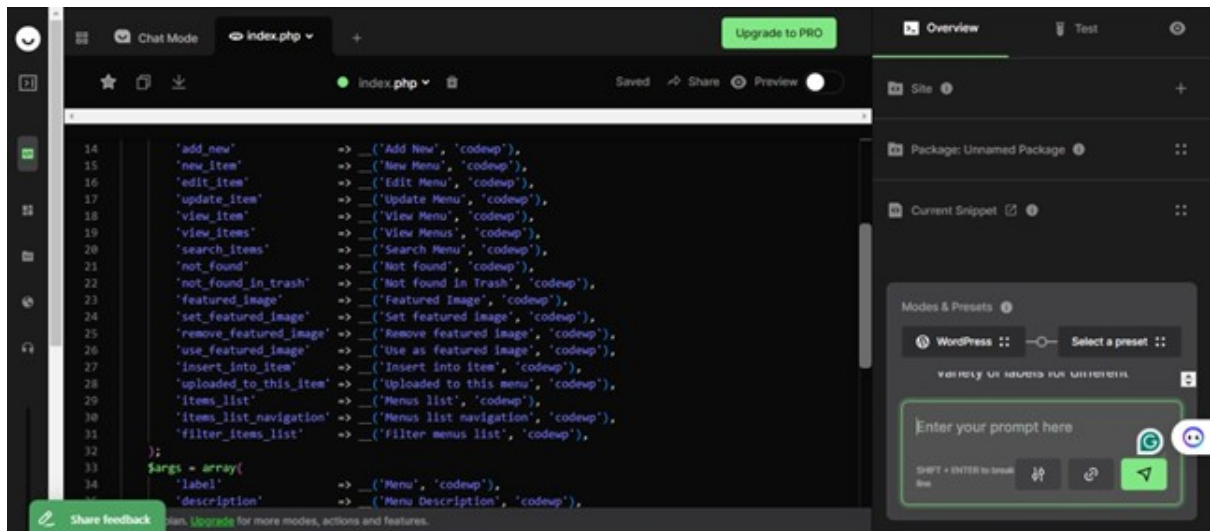


Figure 3: Front end of the CodeWP system

Zyro [5]: Zyro caters to beginners with its drag-and-drop interface. It analyzes user behavior to suggest website templates and content, making the creation process smooth. While this is great for basic websites with common features, extensive customization is limited. Think of Zyro as a website kitset—easy to assemble but with pre-defined options. The front end of the system Zyro is shown in Figure 4.

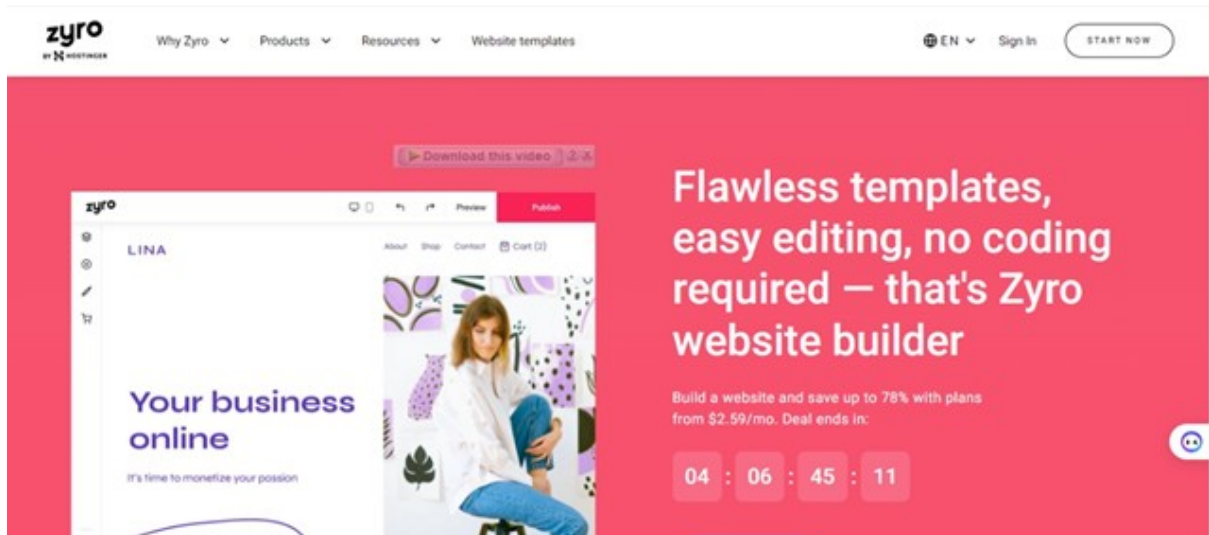


Figure 4: Front end of the Zyro system

Grid [6]: Grid takes a different approach. It leverages AI to generate one-of-a-kind website layouts based on your input and current design trends. This can significantly speed up development and lead to unique websites. However, you might have less control over specific design elements like fonts or color schemes. Imagine Grid as an artistic collaborator who creates a base design for you to refine. The front end of the system Grid is shown in Figure 5.

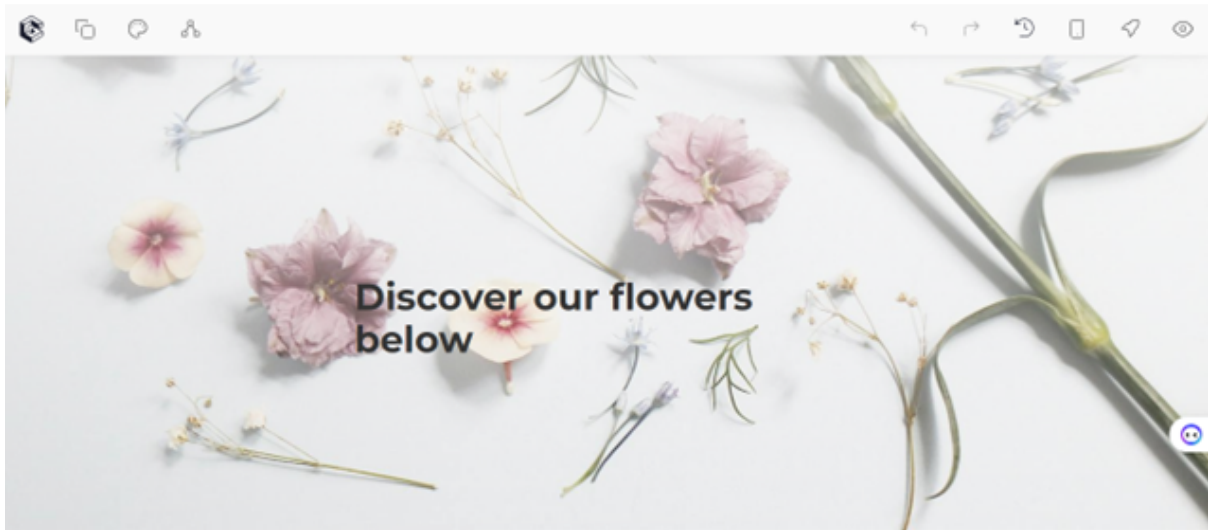


Figure 5: Front end of the Grid system

Webflow [7]: Webflow offers the most power for those with some design knowledge. It utilizes AI to suggest layouts and functionalities based on design patterns and your website data. The real magic lies in the visual interface that allows code-free development. Plus, integration with a content management system makes managing your website a breeze. Think of Webflow as a full-fledged design studio with an AI assistant and a user-friendly content management tool. The front end of the system Webflow is shown in Figure 6.

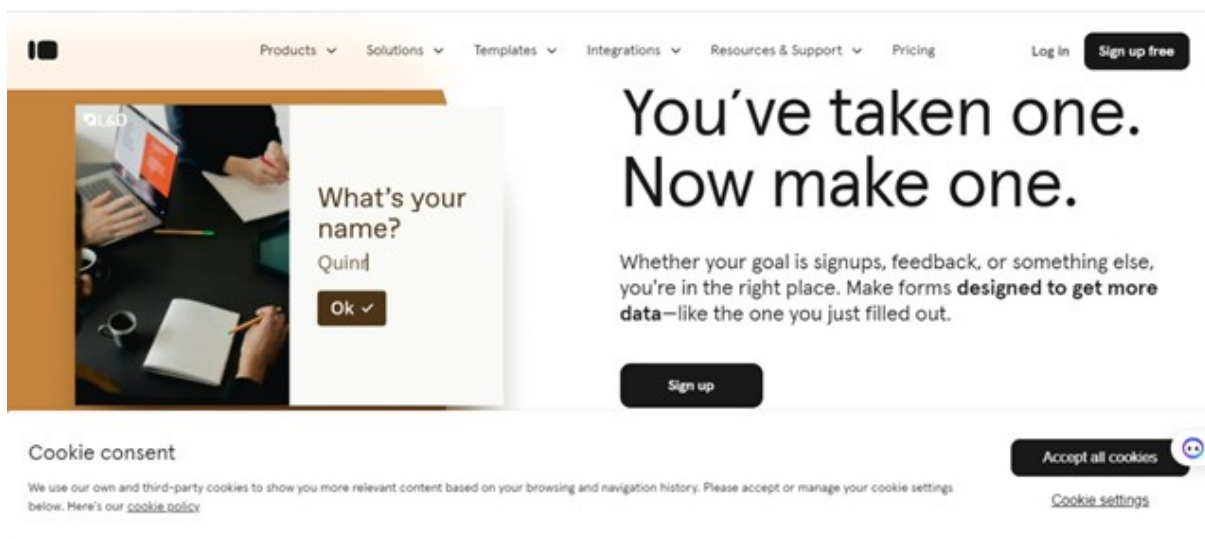


Figure 6: Front end of the Webflow system

Website Name	Dataset	Model and Techniques	Results	Limitations
Wix[?]	Not publicly available	<ul style="list-style-type: none"> - Machine Learning: Classification algorithms to understand user preferences. - Natural Language Processing: Analyze user input to generate website content. 	<ul style="list-style-type: none"> - Ease of use: Creates websites without coding knowledge. - Personalization: Tailors website based on user responses. 	<ul style="list-style-type: none"> - Limited design flexibility: Users have limited control over website aesthetics. - Black box nature: Limited transparency into model decision-making.
Jimdo[?]	Not publicly available	<ul style="list-style-type: none"> - Machine Learning: Similar to Wix ADI, uses classification algorithms. - Natural Language Processing: Analyzes user input to create website content and structure. 	<ul style="list-style-type: none"> - Simple and user-friendly: Ideal for beginners with no coding experience. - Pre-built elements: Offers a variety of pre-designed templates and elements. 	<ul style="list-style-type: none"> - Limited Model: Not much use of NLP and ML. Users have to select options from drop-down. - Limited customization: Offers less flexibility compared to advanced website builders.
Unbounce[?]	Not publicly available, but focuses on conversion	<ul style="list-style-type: none"> - Machine Learning: Utilizes A/B testing and conversion rate optimization algorithms. - Natural Language Processing: Analyzes website copy to improve user engagement. 	<ul style="list-style-type: none"> - Increased conversion rates: Optimizes landing pages for higher conversions. - AI-powered copywriting: Generates persuasive website copy suggestions. 	<ul style="list-style-type: none"> - Specific focus: Primarily targets landing page creation, not general website building. - Limited design options: Offers fewer design elements compared to general website builders.

Table 1: Comparison of Website Builders and AI Integration (Part 1)

Website Name	Dataset	Model and Techniques	Results	Limitations
CodeWP[?]	Not publicly available, but likely uses code generation datasets.	<ul style="list-style-type: none"> - Deep Learning: Employs techniques like neural networks for code generation. - Natural Language Processing: Understands user requirements and translates them into code. 	<ul style="list-style-type: none"> - Efficient development: Automates repetitive coding tasks, saving time. - Complex website creation: Enables building intricate functionalities without manual coding. 	<ul style="list-style-type: none"> - Technical expertise required: Users need basic programming knowledge to effectively utilize the platform. - Limited creative control: Code generation may require adjustments for specific design needs.
Zyro[?]	Zyro uses a drag-and-drop interface with pre-built templates to simplify website creation.	<ul style="list-style-type: none"> - Machine learning is used to analyze user behavior and suggest relevant templates and content. 	<ul style="list-style-type: none"> - Easy to use for beginners, good for basic websites with common features like forms and galleries. 	<ul style="list-style-type: none"> - Limited customization options: Design control is restricted to what pre-built templates offer. May not be suitable for complex or unique website designs.
Grid[?]	Grid focuses on generating unique website layouts using AI.	<ul style="list-style-type: none"> - Generative AI: Creates one-of-a-kind website layouts based on user preferences and current design trends. 	<ul style="list-style-type: none"> - Faster development process with unique layouts. 	<ul style="list-style-type: none"> - Lacks control over specific design elements like fonts or color schemes. The generated layouts might require adjustments and may not always align perfectly with your vision.
Webflow[?]	Webflow offers a powerful visual interface for designing and building websites.	<ul style="list-style-type: none"> - Neural Networks: Analyzes design patterns and user data to suggest layouts and functionalities. - Integrates with a content management system (CMS) for easy content updates. 	<ul style="list-style-type: none"> - Highly flexible design with code-free development. - Integrates with CMS for content management. 	<ul style="list-style-type: none"> - Steeper learning curve compared to Zyro and Grid. Requires some design knowledge and understanding of website structure to use effectively.

Table 2: Comparison of Website Builders and AI Integration (Part 2)

1.9 Team Members Individual Tasks

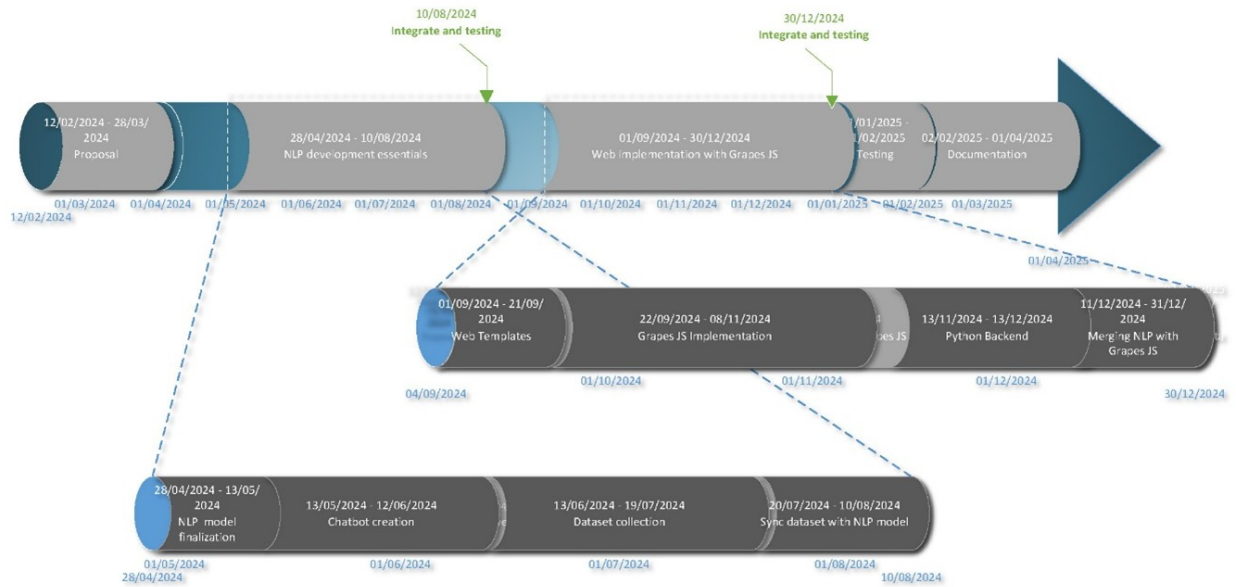
The expected task division among the group members is as given below:

Group Member	Tasks
Naqeeb Ahmed	<ul style="list-style-type: none">- Model Development- Grapes Implementation- Frontend Development- Documentation
Munib Ur Rehman	<ul style="list-style-type: none">- Dataset- Frontend Development- Documentation- Testing
Ahsan Bashir	<ul style="list-style-type: none">- Model Development- Dataset- Backend Development- Testing

Table 3: Team Members and Individual Tasks

1.10 Gantt Chart

A Gantt chart is a visual tool that shows tasks on a timeline. It helps teams see what needs to be done and when, making it easier to manage projects. Based on the project, each task is represented by a horizontal bar, showing its start and end dates along the project timeline. This visual representation helps in understanding the project's schedule and the overlapping of tasks.



ID	Task Name	Start	Finish	Duration	2024												2025		
					Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Jan	Feb	Mar	
1	Project Selection	12/02/2024	22/02/2024	11d	<div></div>														
2	Background Study	23/02/2024	01/03/2024	9d	<div></div>														
3	Proposal	04/03/2024	28/03/2024	24d	<div></div>														
4	NLP model finalization	29/04/2024	13/05/2024	15d	<div></div>														
5	Chatbot creation	13/05/2024	20/06/2024	38d	<div></div>														
6	Dataset collection	21/06/2024	19/07/2024	28d	<div></div>														
7	Sync Dataset with NLP model	22/07/2024	09/08/2024	17d	<div></div>														
8	Website Design	04/09/2024	20/09/2024	17d	<div></div>														
9	Grapes JS Implementation	23/09/2024	08/11/2024	46d	<div></div>														
10	Python Backend	13/11/2024	13/12/2024	31d	<div></div>														
11	Merging of NLP with Grapes JS	16/12/2024	31/12/2024	15d	<div></div>														
12	Testing of Grapes JS	01/01/2025	14/01/2025	15d	<div></div>														
13	Testing of NLP model	14/01/2025	24/01/2025	11d	<div></div>														
14	Overall Testing	24/01/2025	31/01/2025	8d	<div></div>														
15	Documentation	03/02/2025	01/04/2025	57d	<div></div>														

Figure 7: Gantt Chart

1.11 Data Gathering Approach

Data will be gathered through the following methods:

- **Surveys:** Different websites will be surveyed to understand their main points and requirements in managing websites.

- **Analysis of Existing Systems:** Different websites will be analyzed to identify best practices and areas for improvement.
- **Studying Customer Needs:** The team will study customer expectations and preferences to ensure that the system fulfills their requirements.

1.12 References

1. Wix, “Wix.” Accessed: Mar. 29, 2024. [Online]. Available: <https://wix.com/ai-assistant?flow=next>
2. Jimdo, “Jimdo Builder.” Accessed: Mar. 29, 2024. [Online]. Available: <https://cms.jimdo.com/cms/>
3. Unbounce, “Unbounce Web Builder,” [Online]. Available: <https://app.unbounce.com>
4. C. WP, “CodeWP,” [Online]. Available: <https://app.codewp.ai/dashboard/packages>
5. Zyro, “Zyro AI Builder.” Accessed: Mar. 29, 2024. [Online]. Available: <https://zyro.com/>
6. Grid, “Grid AI Web Builder.” Accessed: Mar. 29, 2024. [Online]. Available: <https://use.grid.studio/editor/c5VvkH50SlYy5vsiemg9>
7. TypeForm, “Type Form Webflow.” Accessed: Mar. 29, 2024. [Online]. Available: <https://8ue430n1k59.typeform.com/to/IA2SnQRf>