A PROJECT REPORT

On

Vikram Aur Betaal

Submitted to

KIIT Deemed to be University

In Partial Fulfilment of the Requirement for the Award of

BACHELOR’S DEGREE IN
COMPUTER SCIENCE AND ENGINEERING

BY

ANURAG PANDA	2105181
ANKIT HATI	21052897
PRAJNABRATA MOHANTY	2105049

UNDER THE GUIDANCE OF

MR. N. BIRAJA ISAAC

SCHOOL OF COMPUTER ENGINEERING

KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY

BHUBANESWAR, ODISHA -751024

KIIT Deemed to be University

School of Computer Engineering
Bhubaneswar, ODISHA 751024

CERTIFICATE

This is certified that the project entitled

“VIKRAM & BETAAL”

submitted by

ANURAG PANDA	2105181
ANKIT HATI	21052897
PRAJNABRATA MOHANTY	2105049

is a record of bonafide work carried out by them, in the partial fulfilment of the requirement for the award of Degree of Bachelor of Engineering (Computer Science & Engineering) at KIIT Deemed to be university, Bhubaneswar. This work is done during year 2025, under our guidance.

Project Guide

MR. N BIRAJA ISAAC

Acknowledgements

We are profoundly grateful to **MR. N BIRAJA ISAAC** of **School of Computer Engineering** for his expert guidance and continuous encouragement throughout to see that this project rights its target since its commencement to its completion. …………..

ANURAG PANDA
ANKIT HATI
PRAJNABRATA MOHANTY

ABSTRACT

‘*Vikram and Betaal’* is a classic collection of Indian tales rooted in ancient folklore and moral philosophy. The stories revolve around the legendary King Vikramaditya and the ghostly spirit Betaal, who resides in a corpse hanging from a tree. Each night, as Vikram attempts to capture Betaal and carry him away, the spirit tells him a gripping story that ends with a riddle or moral dilemma. Bound by a condition that he must answer the question if he knows the answer, Vikram repeatedly solves the puzzles, causing Betaal to escape and restart the cycle. These tales are known not just for their rich narrative style, but also for exploring complex themes of justice, duty, wisdom, and human nature. Blending suspense, ethical inquiry, and folklore, ‘*Vikram and Betaal’* remains an enduring part of Indian literature and storytelling tradition.

Keywords: Vikramaditya, Betaal, Indian folklore, moral stories, riddles, ethical dilemmas, ancient literature, wisdom, justice, storytelling.

Contents

n

List of Figures:

Chapter 1

Introduction

Vikram and Betaal is one of the most iconic story cycles in Indian literature, blending elements of fantasy, morality, and suspense. Originating from the ancient Sanskrit work *Baital Pachisi*, attributed to the scholar Somdev Bhatt, these tales date back over a thousand years and have been retold across generations in various regional languages and cultural forms. The central characters are King Vikramaditya, a just and fearless ruler known for his wisdom and bravery, and Betaal, a witty and mysterious ghost who inhabits a corpse hanging from a tree in a dense forest.

The premise of the stories is simple yet captivating. Vikramaditya is tasked by a tantric sage to retrieve Betaal, but every time the king tries to carry the ghost, Betaal tells him a strange and thought-provoking story that ends with a riddle or a moral dilemma. If Vikram knows the answer and remains silent, he risks death; if he answers, Betaal escapes and returns to the tree, forcing Vikram to start over. This recurring cycle forms the narrative framework of the series.

Each of Betaal’s stories presents a unique scenario that explores complex human emotions, ethical conflicts, and social values. From tales of loyalty and betrayal to stories about justice, sacrifice, and cleverness, the narrative structure encourages the listener or reader to reflect deeply on right and wrong. Despite being rooted in folklore, the themes remain relevant in modern times and offer timeless wisdom.

The *Vikram and Betaal* tales have been widely popularized through books, television series, plays, and even comic books, making them accessible to both children and adults. They not only serve as entertainment but also act as a cultural and moral compass, emphasizing the importance of intellect, integrity, and courage.

This project explores the origins, structure, major themes, and cultural significance of the *Vikram and Betaal* stories, shedding light on why they continue to captivate minds even today.

Chapter 2

Basic Concepts/ Literature Review

This section contains the basic concepts about the related tools and techniques used in this project.

2.1 HTML

HTML (Hypertext Markup Language) is the foundational language used to create and structure content on the web. It serves as the backbone of all websites, providing the essential framework that web browsers interpret and render into viewable pages. Without HTML, there would be no coherent way to organize text, images, links, or multimedia elements on the Internet.

HTML is not a programming language; rather, it is a **markup language** that uses a system of tags and elements to define the content and structure of a web page. These tags describe the role of various content types—such as headings, paragraphs, lists, links, images, and tables—giving web browsers clear instructions on how each element should be displayed.

HTML typically works in conjunction with two key technologies:

- **CSS (Cascading Style Sheets)** – controls the visual style and layout, including colors, fonts, spacing, and responsiveness.
- **JavaScript** – enables interactive behaviors and dynamic content changes, such as form validation or animated elements.

Web browsers retrieve HTML documents either from remote servers (over the internet) or local storage, and then render them into user-friendly interfaces. During this rendering process, the browser parses the HTML to create a **Document Object Model (DOM)**—a structured representation of the page that allows scripts and styles to modify it in real time.

Another important feature of HTML is its **semantic structure**. Modern HTML supports semantic tags like <header>, <footer>, <section>, and <article>, which help define the role and meaning of each content block. This not only improves accessibility for screen readers but also enhances SEO (Search Engine Optimization) and maintainability.

2.1 CSS

CSS (Cascading Style Sheets) is a core technology used in web development that controls the presentation and layout of HTML content. While HTML defines the structure and content of a webpage, CSS determines how that content looks—its colors, fonts, spacing, alignment, animations, and responsiveness across different devices.

CSS works by selecting HTML elements and applying style rules to them. These rules define visual characteristics such as:

- **Typography**: font families, sizes, weights, and line spacing.
- **Color and Backgrounds**: text color, background color or images, gradients, and transparency.
- **Layout and Positioning**: margins, padding, borders, grid systems, and flexbox layout structures.
- **Responsiveness**: media queries allow styles to adapt based on screen size, enabling mobile-friendly designs.
- **Visual Effects**: transitions, transformations, animations, and hover effects enhance user interactivity.

There are three main ways to apply CSS:

1. **Inline styles**: defined directly within HTML elements.
2. **Internal styles**: placed in a <style> tag within the HTML file.
3. **External stylesheets**: linked via a separate .css file, promoting modular and maintainable code.

One of CSS's most powerful features is the **"cascade"**, which determines how conflicting style rules are applied based on **specificity**, **inheritance**, and **source order**. This layered approach gives developers fine-grained control over page styling.

Modern CSS supports advanced features like **CSS Grid** and **Flexbox**, which simplify complex layouts without heavy reliance on JavaScript or external libraries. Additionally, variables, custom properties, and preprocessor tools like Sass have expanded what's possible with CSS, allowing for dynamic and scalable styling systems.

In summary, CSS transforms a basic HTML document into a polished, user-friendly interface. It is essential for creating attractive, consistent, and accessible websites across all platforms and devices.

2.2 BOOTSTRAP

Bootstrap is a powerful, open-source front-end framework designed to help developers quickly build responsive and visually consistent websites and web applications. Originally developed by Twitter, Bootstrap simplifies web design by providing a collection of ready-to-use components, layout utilities, and a responsive grid system—all based on HTML, CSS, and JavaScript.

Key Features:

- **Responsive Grid System**: At the heart of Bootstrap is a 12-column, responsive grid layout that automatically adjusts content for different screen sizes (phones, tablets, desktops).
- **Pre-built Components**: It comes with a wide array of reusable UI components like navigation bars, buttons, forms, cards, modals, carousels, and more.
- **Utility Classes**: Bootstrap provides a rich set of utility classes (e.g., for margins, padding, text alignment, flex layout) that make rapid prototyping easy.
- **Customizable Themes**: Developers can override default styles or use Sass variables to tailor Bootstrap's look to their branding needs.

- **JavaScript Plugins**: Bootstrap includes ready-to-use JavaScript components like dropdowns, tooltips, accordions, and modals, powered by Popper.js and jQuery (or in newer versions, vanilla JavaScript).

In summary, Bootstrap is ideal for developers who want to create responsive and elegant websites quickly without deep diving into custom CSS from scratch. Its ease of use and extensive documentation make it a go-to framework for beginners and professionals alike.

Once included in a project, Bootstrap provides basic style definitions for all HTML components. The ultimate effect is a uniform display of text, tables, and form components across all web browsers. Developers may also use Bootstrap's CSS classes to further customise the look of their content. For example, Bootstrap has provisioned for light- and dark-colored tables, page headings, more prominent pull quotes, and text with a highlight.

2.3 JAVASCRIPT

JavaScript is a high-level, dynamic scripting language that enables interactive and dynamic behavior on websites. It is one of the **core technologies of the web**, alongside HTML and CSS, and is supported natively by all modern web browsers without the need for additional plugins.
Originally created to add interactivity to web pages (like button clicks or form validation), JavaScript has evolved into a full-fledged programming language capable of powering everything from simple animations to complex web applications and servers.

Features:-

- **Event-driven**: Executes code in response to user interactions.
- **Asynchronous programming**: Using Promises, async/await, and callbacks for non-blocking operations (e.g., fetching data from an API).
- **Lightweight and versatile**: Can be embedded in HTML or linked via external .js files.

JavaScript is not limited to web browsers. With platforms like **Node.js**, it can run on servers, handle file systems, interact with databases, and build entire backends.

JavaScript is also a fully-fledged programming language that supports variables, conditional logic, loops, functions, and even object-oriented and functional programming paradigms. Over the years, the language has evolved significantly through updates to the ECMAScript specification. Modern JavaScript, often referred to as ES6 and beyond, introduces syntax and features like arrow functions, template literals, classes, modules, de-structuring, and more—making the language more powerful, readable, and maintainable.

Beyond the browser, JavaScript's reach extends to the server side, thanks to environments like Node.js. With Node, JavaScript can be used to build APIs, handle databases, process files, and serve entire web applications, all from a single codebase. This unification of front-end and back-end development streamlines workflows and reduces the need for multiple programming languages within a single project.

The JavaScript ecosystem is vast and continuously growing. Libraries such as jQuery, and more recently, frameworks like React, Angular, and Vue, have revolutionized the way developers build complex user interfaces. Tools like Webpack and Babel enhance the development process by allowing developers to bundle, transpile, and optimize their JavaScript code for production environments. Additionally, the npm (Node Package Manager) ecosystem offers access to hundreds of thousands of reusable packages that speed up development and reduce redundancy.

In summary, JavaScript is far more than just a scripting language for browsers—it is a powerful, flexible, and ubiquitous tool that underpins much of today's digital world. Whether you are creating simple websit interactions or architecting full-scale web applications, JavaScript remains the language that brings interactivity and intelligence to the web.

2.4 PYTHON

Python is a high-level, interpreted programming language renowned for its readability, simplicity, and versatility. Created by Guido van Rossum and first released in 1991, Python was designed with the philosophy that code should be easy to understand and write. Its clean syntax—often likened to plain English—makes it an ideal choice for beginners, while its depth and extensibility make it equally powerful for professionals working on complex systems.

One of Python's most notable strengths is its multi-paradigm nature. It supports procedural, object-oriented, and functional programming styles, allowing developers to choose the best approach for the problem at hand. Unlike many programming languages that prioritize performance at the cost of readability, Python emphasizes code clarity, which leads to better collaboration, easier maintenance, and faster development cycles. Its dynamic typing and automatic memory management further contribute to its ease of use and flexibility.

Another major advantage of Python is its cross-platform compatibility. It runs on all major operating systems—Windows, macOS, and Linux—and integrates easily with other languages and technologies. This makes Python an excellent choice for building both standalone software and web-based solutions. The language's popularity is further reflected in educational institutions, where it is often chosen as the first language to teach programming, thanks to its low barrier to entry and extensive educational resources.

Python is open-source and continuously improved by a global community of developers. The Python Software Foundation oversees its development and ensures that the language remains stable, secure, and adaptable to modern computing challenges. With frequent updates and enhancements introduced through PEPs (Python Enhancement Proposals), Python has managed to stay relevant and modern while retaining the simplicity that made it famous.

In conclusion, Python is more than just a programming language—it's a gateway to building almost anything in the digital world. Whether you're automating daily tasks, developing complex algorithms, analyzing big data, or teaching a beginner how to code, Python stands out as a language that balances accessibility with immense power.

2.6 FLASK

Flask is a lightweight web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries, offering simplicity and flexibility for developers. Flask includes built-in support for routing, templating via Jinja2, and a WSGI server through Werkzeug.

Flask is commonly used for building RESTful APIs and small to medium-sized web applications. Its modular design allows developers to add extensions for tasks like database integration, authentication, and form validation. While Flask provides a minimal starting point, it is scalable and well-suited for projects requiring fine-grained control over application architecture.

2.7 FAST API

FastAPI is a modern, high-performance web framework for building APIs with Python 3.7+ based on standard Python type hints. Designed with speed, efficiency, and developer experience in mind, FastAPI stands out as one of the fastest Python frameworks available today, rivaling the performance of Node.js and Go in many benchmarks. Its name reflects its focus: creating **Fast APIs**—not only in runtime, but also in development speed.

Built atop **Starlette** for the web layer and **Pydantic** for data validation, FastAPI provides a robust foundation for building scalable, asynchronous APIs with minimal boilerplate code. The framework leverages Python's type annotations to automatically generate documentation, validate request and response bodies, and improve editor support with features like autocomplete and type checking. This tight integration with type hints results in highly readable, self-documenting code and dramatically reduces the chances of runtime errors.

One of FastAPI's most compelling features is its automatic generation of interactive API documentation using **OpenAPI** and **Swagger UI**. As soon as an endpoint is defined, developers can test and explore the API through a web-based interface, making development and debugging faster and more intuitive. This built-in documentation also greatly aids collaboration between backend and frontend developers or API consumers.

FastAPI is fully compatible with **asynchronous programming**, allowing it to handle thousands of concurrent requests with efficient use of resources. This makes it particularly well-suited for building high-throughput services such as microservices, real-time APIs, or applications that need to integrate with external services asynchronously. Despite its focus on performance, FastAPI doesn't sacrifice clarity or simplicity, offering an elegant developer experience that scales with the complexity of the project.

The framework integrates easily with databases, ORMs like SQLAlchemy or Tortoise ORM, and background task processing tools such as Celery. It is commonly used in production by teams building data pipelines, backend APIs for mobile and web apps, and machine learning model deployment endpoints. Its growing popularity is also due to its strong alignment with modern Pythonic principles and community-driven development.

In summary, FastAPI combines the **power of asynchronous execution**, the **clarity of type hints**, and the **automation of modern documentation tools** into a single, easy-to-use framework. It allows developers to build robust, production-ready APIs with less code, fewer bugs, and greater productivity—making it an ideal choice for both rapid prototyping and large-scale applications.

2.8 SQLite

SQLite is a compact, self-contained relational database engine that stores data in a single file, making it incredibly easy to set up and use. Unlike traditional database systems that require a server, SQLite runs within the application itself, meaning there’s no need for installation, configuration, or a network connection. This simplicity makes it ideal for mobile apps, embedded systems, desktop applications, and small-scale web projects.

Despite its lightweight nature, SQLite supports most of the SQL standard, including joins, indexes, transactions, and subqueries. It ensures data integrity through full ACID compliance, meaning operations are safe, consistent, and reliable—even during crashes or power failures. Its file-based design allows easy portability, backups, and sharing between environments or platforms.

While SQLite isn't built for high-concurrency, server-scale workloads, it shines in situations where low overhead, minimal setup, and ease of use are key. Developers often choose it for prototyping, local storage, or applications where a full server-based database would be overkill. In short, SQLite delivers robust database capabilities with a fraction of the complexity—perfect for projects that value simplicity and efficiency.

Chapter 3

Requirement Specifications

It covers functional and non-functional requirements, as well as the technical infrastructure necessary to ensure high performance, reliability, security, and maintainability.

Functional Requirements:

User Interface and Design:

The website must feature a modern, user-friendly interface with a responsive design that adapts to desktops, tablets, and mobile devices. Standard pages should include:

- Home
- About Us
- Comics
- Contact Us
- Chatbot

All pages should have a consistent layout, branding elements (such as logo and color scheme), and clear navigation.

Navigation and Structure:

- A global navigation bar should be present on all pages.
- Mobile versions must feature a collapsible or hamburger-style menu.
- Internal links must be properly structured for smooth navigation.

Forms and User Input:

- The Contact Us page must include a form with Name, Email, Subject, and Message fields.
- All input fields should have client-side and server-side validation.
- Form submissions must trigger a visual confirmation message.
-

Content Display:

- Text, images, and videos must load properly and be formatted responsively.
- Content should be editable through a CMS (if dynamic content is required).
- Optional sections such as Recent News, Portfolio, or Blog Posts may be included.

Non-Functional Requirements:

Performance:

- Pages must load within 3 seconds on a standard broadband connection.
- Assets such as images and scripts should be optimized and compressed.
- Use of caching and lazy loading to improve performance.

Security:

- HTTPS must be enforced through an SSL certificate.
- User inputs should be sanitized to prevent XSS or injection attacks.
- Admin and CMS access must be secured through authentication.
- Data transmission and storage (if applicable) must comply with data protection standards.

Reliability and Uptime

- The website should maintain an uptime of 99.9%.
- A custom 404 page should be implemented for handling broken links.
- Regular backups should be taken and stored securely.

Usability and Accessibility

- The design must follow WCAG 2.1 accessibility standards.
- Users should be able to navigate via keyboard and screen readers.
- Interactive elements should provide visual feedback (e.g., hover states, focus outlines).

Compatibility

- The website must function consistently across all major browsers: Chrome, Firefox, Safari, and Edge.
- It should render correctly across different screen resolutions.

Technical Requirements:

Frontend Technologies:

- HTML5: for content structure
- CSS3 / Bootstrap: for styling and layout
- JavaScript: for interactivity
- Optional: Use of libraries like jQuery, or frameworks like React or Vue.js

Backend Technologies:

- Python (FastAPI or Flask), Node.js, or PHP for server-side processing
- SQLite for lightweight data storage or MySQL/PostgreSQL for scalable data requirements
- Email server or API integration for handling contact form submissions

Hosting and Deployment:

- A domain name with a registered SSL certificate
- Hosting solution (Shared, VPS, or Cloud-based) with at least 99.9% uptime guarantee
- FTP/SFTP access or CI/CD pipeline for deployment

Maintenance and Support:

- Regular updates for software and dependencies
- Security patches applied as needed
- Monthly monitoring of uptime, broken links, and analytics
- Documentation for managing content updates

Chapter 4

Concepts Used

MACHINE LEARNING

Machine Learning (ML) is a powerful branch of **artificial intelligence** that enables computer systems to autonomously learn from data, recognize underlying patterns, and make decisions or predictions with minimal human intervention. It is particularly valuable in applications involving natural language processing (NLP), where the goal is to understand and generate human-like language. In this project, ML was utilized to develop a chatbot capable of interpreting user emotions and providing meaningful responses rooted in Indian philosophy and storytelling.

The core idea was to create a chatbot that could emotionally engage with users by offering them personalized spiritual and moral guidance. To do this, we trained a model on a carefully curated dataset containing Bhagavad Gita verses and Vikram and Betaal stories, which are rich in moral, philosophical, and emotional depth. These texts were pre-processed through multiple stages, including tokenization, stop word removal, stemming or lemmatization, and vectorization, to convert raw textual data into structured formats suitable for machine learning algorithms.

The chatbot uses a basic sentiment analysis or emotion detection technique to first classify the user's emotional input—such as stress, sadness, confusion, or curiosity. Based on this classification, the model matches the user's state to thematically relevant verses from the Bhagavad Gita that offer wisdom and reassurance. Simultaneously, it recommends a Vikram and Betaal story that echoes the same moral or philosophical message, providing a layered and culturally resonant response.

Technologies such as Python were instrumental in building and integrating this ML pipeline. The model is seamlessly embedded into the backend of the website, allowing real-time interaction and response generation without manual intervention.

Through this approach, machine learning not only adds intelligent behavior to the system but also plays a transformative role in connecting users with ancient wisdom in a personalized, modern, and meaningful way. This fusion of culture, emotion, and AI demonstrates the potential of technology to serve as a bridge between tradition and innovation.

Using **Natural Language Processing (NLP)** techniques, the machine learning model is capable of interpreting user input with a degree of linguistic and emotional understanding. When a user submits a query expressing their current feelings or seeking guidance, the model processes the input to identify key emotional cues, themes, and intent. This is achieved through techniques such as tokenization, part-of-speech tagging, named entity recognition, and semantic similarity matching, allowing the system to grasp not just the literal meaning of the message but also its deeper context.

To build this functionality, we employed supervised learning, training the model on a labeled dataset where various emotional states and query patterns were matched with corresponding Bhagavad Gita verses and Vikram and Betaal stories. Through multiple training iterations, the model learned to associate specific emotional triggers and philosophical themes with appropriate responses. This resulted in a chatbot that can provide contextually accurate and spiritually meaningful answers, tailored to the user’s state of mind.

The integration of this trained ML model into the website's backend—using Python and FastAPI—enables real-time interaction. When a user types a message, the backend processes it through the NLP pipeline, generates a suitable response from the Gita and story database, and returns the result instantly to the user interface. This creates a smooth, engaging user experience that blends ancient wisdom with modern AI technology.

This implementation not only showcases the practical application of machine learning in NLP but also illustrates how spiritual and philosophical texts can be transformed into a dynamic, interactive medium. It makes deep teachings more accessible, relatable, and engaging, especially for younger audiences exploring these ideas in a digital space.

SENTIMENTAL ANALYSIS

Sentiment Analysis is a Machine Learning (ML) technique that interprets the emotional tone of text to classify it into categories like positive, negative, or neutral sentiments. In this project, sentiment analysis was implemented to enable the chatbot to understand the emotional state of users based on their inputs and provide relevant morals and stories with respect to Bhagwat Geeta and Vikram and Betaal.

The chatbot analyzes user inputs by identifying specific keywords and expressions indicative of emotions such as anger, sadness, happiness, or anxiety. A curated dataset mapping emotional expressions to corresponding shlokas was used to train the ML model. By leveraging **Natural Language Processing (NLP)** techniques, the model extracts features from the user input, determines the user's mood, and retrieves appropriate shlokas that align with the identified sentiment.

For example, if a user types, "I’m feeling angry," the chatbot detects the emotion as anger and suggests shlokas that provide guidance on managing anger. This integration of sentiment analysis ensures that the chatbot delivers a personalized and meaningful experience, making the teachings of the Bhagavad Gita accessible and applicable to users' emotional contexts.

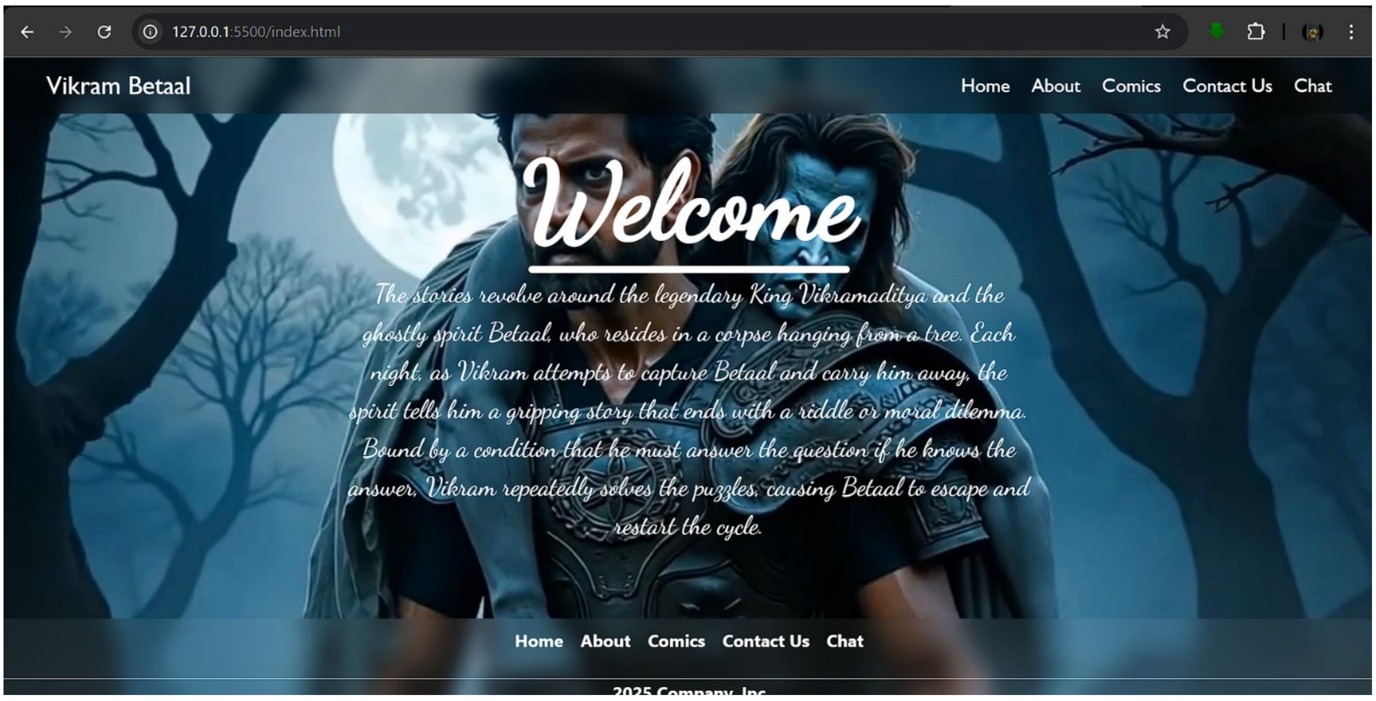
Chapter 5

Implementation

5.1 LAYOUT

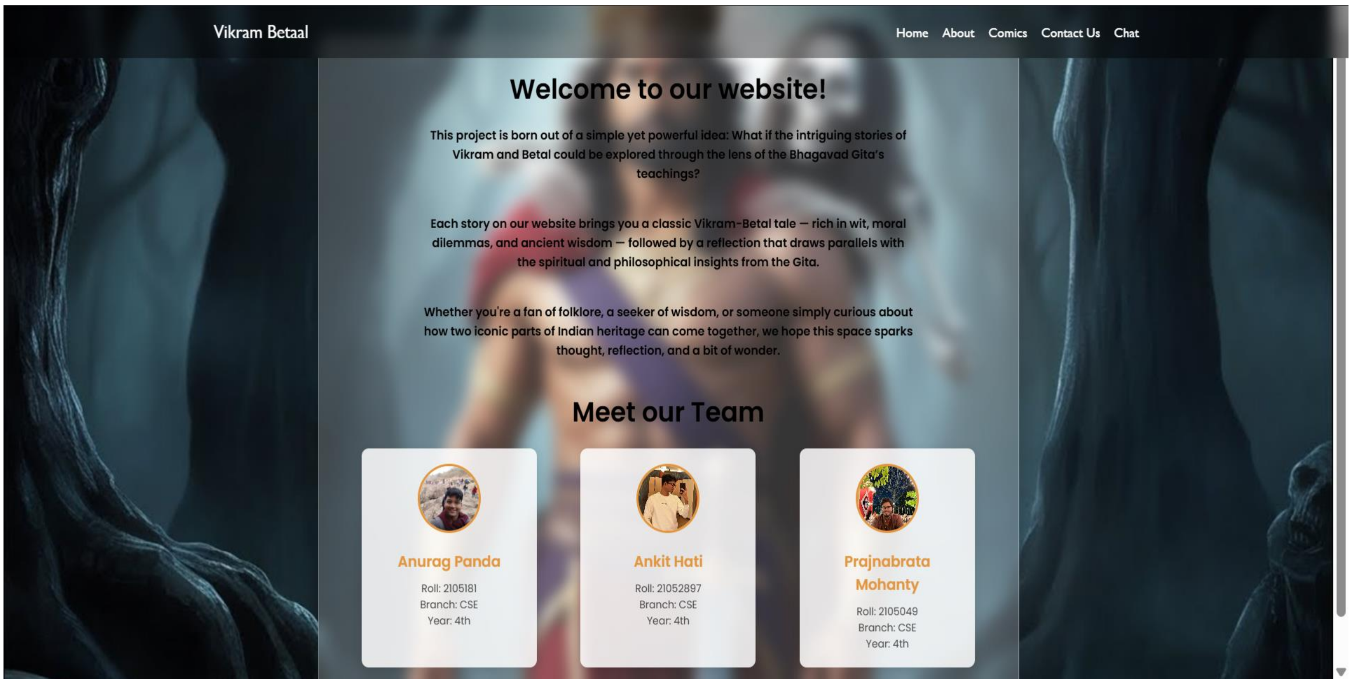
The homepage serves as both the initial page visitors see and a landing page that encourages a specific action. In this case, the 'Cantos' link is the main call to action, distinguished by a uniquely colored border to make it stand out from other links and guide users' decisions. Although other links are present, the call to action is strategically placed beneath a key element and again at the bottom of the page, increasing its visibility. Based on the Gutenberg Diagram, this call to action sits in the terminal area—where users typically make decisions—while the navigation bar occupies the primary optical area, where attention naturally falls. This helps users quickly understand the site’s purpose and easily access other pages. A parallax scrolling design was chosen to keep the site modern and engaging. It appeals to younger users familiar with scrolling interfaces on mobile devices, offering a more intuitive and interactive experience.

Home Page:



The stories revolve around the legendary King Vikramaditya and the ghostly spirit Betaal, who resides in a corpse hanging from a tree. Each night, as Vikram attempts to capture Betaal and carry him away, the spirit tells him a gripping story that ends with a riddle or moral dilemma. Bound by a condition that he must answer the question if he knows the answer, Vikram repeatedly solves the puzzles, causing Betaal to escape and restart the cycle.

About Us Page:



Welcome to our website!

This project is born out of a simple yet powerful idea: What if the intriguing stories of Vikram and Betal could be explored through the lens of the Bhagavad Gita's teachings?

Each story on our website brings you a classic Vikram-Betal tale — rich in wit, moral dilemmas, and ancient wisdom — followed by a reflection that draws parallels with the spiritual and philosophical insights from the Gita.

Whether you're a fan of folklore, a seeker of wisdom, or someone simply curious about how two iconic parts of Indian heritage can come together, we hope this space sparks thought, reflection, and a bit of wonder.

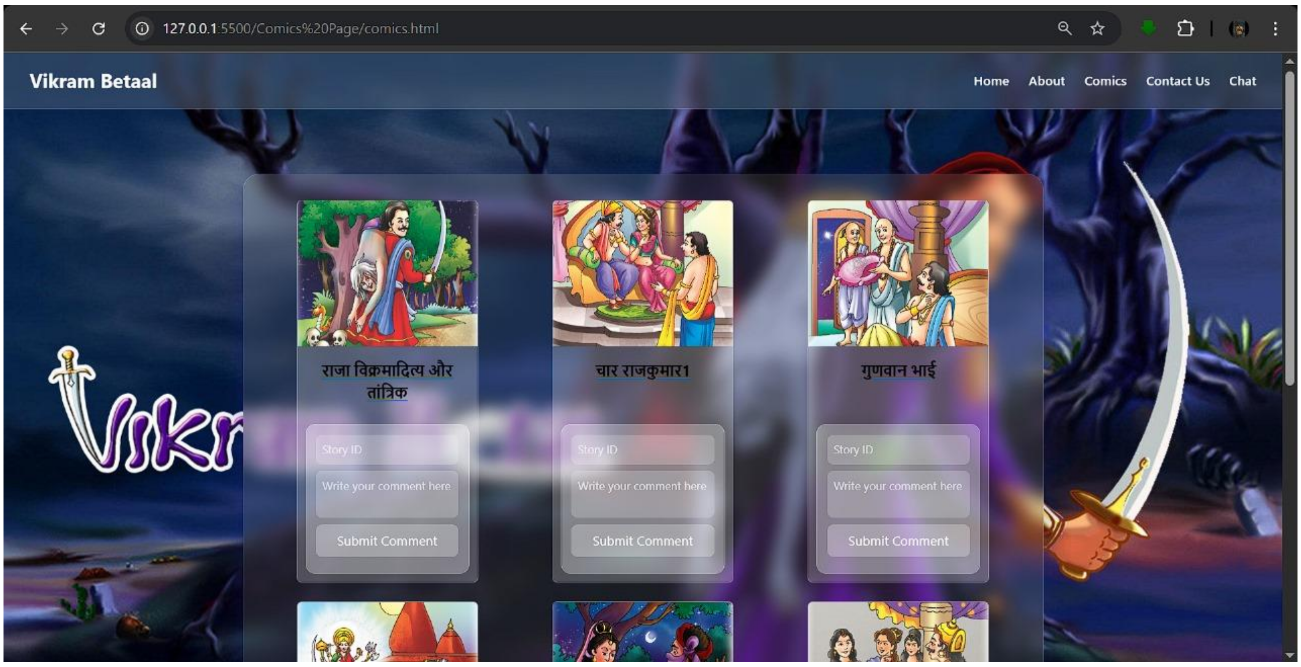
Meet our Team

Anurag Panda
Roll: 2105181
Branch: CSE
Year: 4th

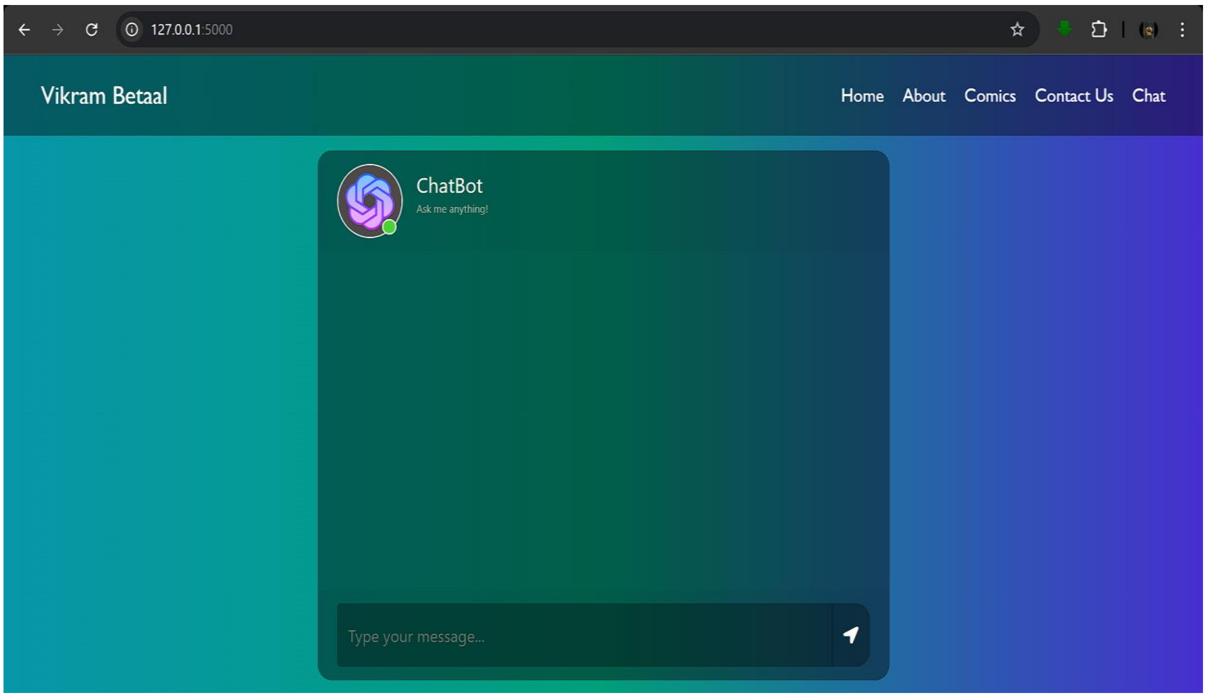
Ankit Hati
Roll: 21052897
Branch: CSE
Year: 4th

Prajnabrata Mohanty
Roll: 2105049
Branch: CSE
Year: 4th

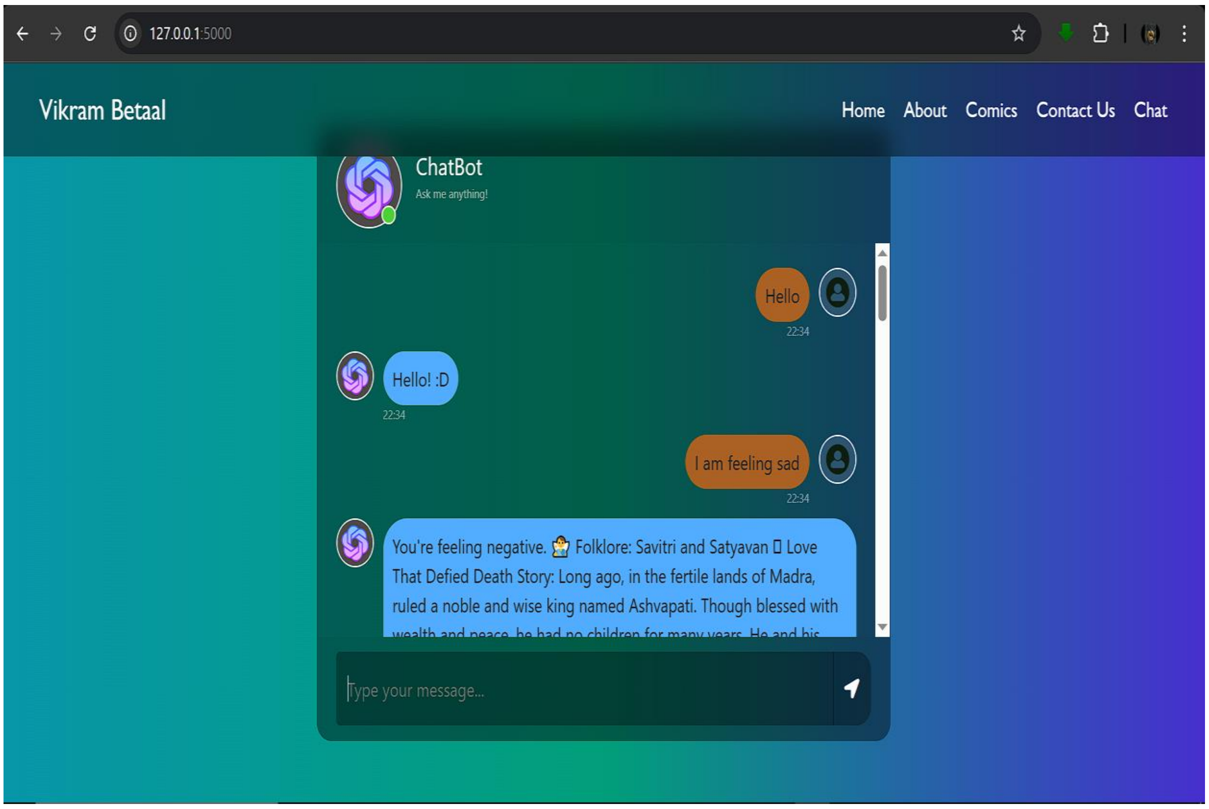
Comics Page:



Chatbot:



Chatbot (while being used):



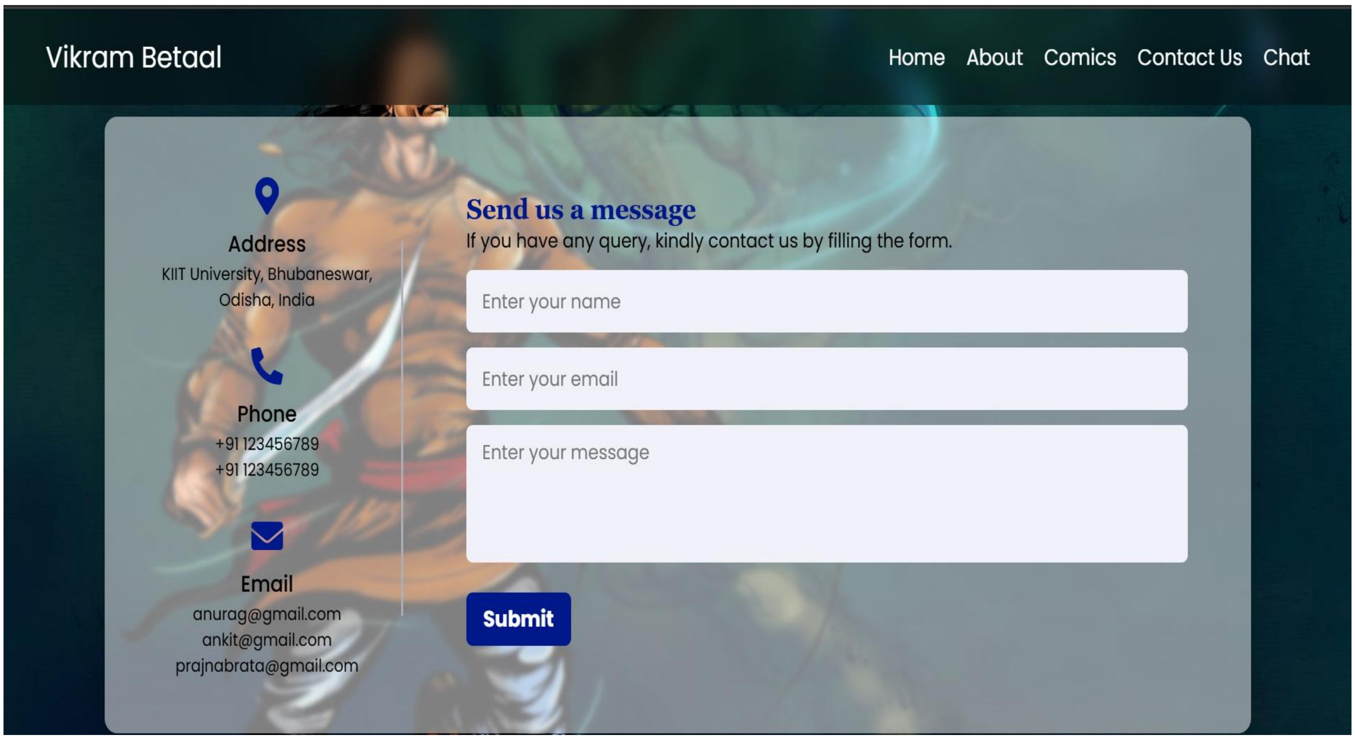
Hello
22:34

Hello! :D
22:34

I am feeling sad
22:34

You're feeling negative. 🧚 Folklore: Savitri and Satyavan ▯ Love That Defied Death Story: Long ago, in the fertile lands of Madra, ruled a noble and wise king named Ashvapati. Though blessed with wealth and peace, he had no children for many years. He and his

Contact Us Page:


Vikram Betaal

Home About Comics Contact Us Chat

Address

KIIT University, Bhubaneswar, Odisha, India

Phone

+91 123456789
+91 123456789

Email

anurag@gmail.com
ankit@gmail.com
prajnabrata@gmail.com

Send us a message

If you have any query, kindly contact us by filling the form.

Enter your name

Enter your email

Enter your message

Submit

5.2 MOBILE OPTIMIZATION

To increase the amount of prospective audience members, this site employs media queries. This website is responsive to three distinct sizes of devices:

- Smartphones >500px
- Tablets >960px
- Desktop / Laptops <960px

According to Kemp (2017), more than half of the world's population now has a smartphone, and smartphones account for more than half of global web traffic, making it critical that a website be mobile-friendly. However, not only are sites being accessed by more people on more devices, “customers are returning to our sites at different times using different devices.” Therefore, having a responsive website is a key trend that modern websites must incorporate. To minimize loading times on smaller devices, the background video changes to a still image on devices under 960px.

The call to action adapts to have a background color to keep the user’s attention as there is no ability to hover a mouse on anything other than desktops or laptops.

5.3 COLORS

Colour has a significant impact on the entire appearance and feel of a website, and it is typically the first impression visitors have of you, deciding whether they want to remain or go. Colors have different meanings in different cultures and countries, so understanding the meaning of color in your target market can be important. The color palette used for the website was kept really clean and simple.

The colors used for the website:
white

- rgba (0, 0, 0, 0.6) [BLACK/ 60% opacity]
- rgba (255, 255, 255, 0.2) [WHITE/ 20% opacity]
- rgba (0, 0, 0, 0.25) [BLACK/ 25% opacity]
- rgba (255, 255, 255, 0.3) [WHITE/ 30% opacity]
- #000000 [BLACK]
- rgba (255, 255, 255, 0.85) [WHITE/ 85% opacity]
- rgba (0,0,0,0.1) [BLACK/ 10% opacity]
- #e3963e [TIGER ORANGE]

Background colors used in the Chatbot:
Gradient 1: -webkit-linear-gradient (rgb(40, 59, 34), rgb(70, 61, 54), rgb(32, 32, 43)).
Gradient 2: Linear-gradient (rgb(168, 146, 37), rgb(255, 131, 22), rgb(146, 46, 37)).
White for clean sections.
Transparent Overlay: rgba(0,0,0,0.3).

5.4 FONTS

To avoid looking messy, the best is to use a maximum of two or three fonts on a page. For the main font, the one used for the title, we wanted something modern but again not too distracting.

The fonts used here:
1. 'Poppins', sans-serif
2. Regular

All the fonts used are used within a list of fallback fonts because not every computer or every browser will have the same fonts available. In this case, if the first font in the list is not available, the browser will try to use the next font specified, and so on.

5.5 Workflow Pipeline:

The development of our interactive web platform followed a structured workflow designed to ensure robust integration of frontend design, backend services, and machine learning components. The goal was to create an emotionally intelligent system that bridges Indian mythology with spiritual wisdom through engaging user interaction.

- **Requirement Analysis and Planning**

  The initial phase involved defining the project's objectives: to create an engaging website centred around Vikram and Betaal stories, combined with a mood-aware chatbot that recommends relevant Bhagavad Gita verses and moral tales. This phase also included selecting appropriate technologies such as HTML, CSS, JavaScript for the frontend, and Python with FastAPI for the backend.

- **Frontend Development**

  The user interface was crafted using HTML and CSS, enhanced with a glass morphism design for a modern aesthetic. JavaScript was employed to handle dynamic behaviours and facilitate communication with the backend API. Key pages like *About Us* and *Contact Us* were developed to provide context and gather user feedback.

- **Dataset Collection and Processing**

  Texts from the Bhagavad Gita and Vikram-Betaal stories were compiled and labelled based on their emotional or moral themes. Natural Language Processing (NLP) techniques such as tokenization, stop word removal, and text normalization were applied to prepare the data for machine learning model training.

- **ML Model Development**

  Supervised learning algorithms, including Decision Tree, Support Vector Machine (SVM), and Random Forest, were trained on the pre-processed dataset to classify user moods and map them to appropriate Gita verses and story morals. The trained models were evaluated using metrics like accuracy, precision, recall, and F1-score to select the most effective one for deployment.

- **Backend Development with FastAPI**

  The backend was developed using FastAPI, a modern Python web framework known for its speed and simplicity. It handled:

  - API endpoints to receive user queries.
  - Integration with the ML model to process inputs and return predictions.
  - Serving the relevant Gita verse and story.
  - Connection to a SQLite database for storing and managing story and verse data.

- **Integration and Testing**

  Once both frontend and backend components were ready, they were integrated through RESTful APIs. Comprehensive testing was conducted to ensure accurate data flow, responsive design across devices, and robust error handling. This phase also included validating the chatbot's performance in delivering contextually appropriate responses.

- **Project Documentation**

  Detailed documentation was prepared to encapsulate the project's development process, including system architecture diagrams, code snippets, and user guides. This documentation serves as a valuable resource for future maintenance and potential scalability enhancements.

- **Deployment and Demonstration**

  The final system was deployed locally and demonstrated.

  The deployment showcased the seamless interaction between users and the chatbot, highlighting the system's capability to provide personalized spiritual and moral guidance based on user input.

Chapter 6

Conclusion

The goal of this project is to make the timeless teachings of the Bhagavad Gita and the moral wisdom found in Vikram and Betaal accessible and relevant to today's generation. Many young individuals remain unaware of the deep philosophical insights embedded within these ancient texts, which address universal themes such as duty, fear, self-inquiry, and moral choice.

By leveraging Machine Learning and Sentiment Analysis, this project introduces a chatbot that recommends personalized shlokas from the Gita and stories or riddles inspired by Vikram and Betaal, based on the user's emotional state. Whether a user is feeling anxious, lost, curious, or reflective, the chatbot offers contextual guidance—helping them face inner conflicts, ethical dilemmas, and moments of doubt with clarity and strength.

Just as Krishna guides Arjuna to embrace his dharma with courage, and Betaal challenges Vikram with questions that demand wisdom and introspection, this project uses modern technology to simulate that same journey of self-awareness

and moral reasoning. Ultimately, the aim is to foster courage, discernment, and inner growth while reconnecting users with the rich spiritual and intellectual

heritage of Indian storytelling and philosophy**.**

Future Scope:

The current version of the project successfully integrates storytelling and spiritual guidance through a chatbot that uses machine learning and natural language processing. However, there is significant potential for expanding and enhancing the platform in the future to create an even more immersive and impactful experience for users.

One promising direction is the integration of voice-based interaction, allowing users to speak to the chatbot rather than type. This would increase accessibility, especially for younger users and those less familiar with typing or digital interfaces. Incorporating speech recognition and text-to-speech (TTS) systems would make the experience more natural and engaging.

Additionally, the chatbot’s emotional intelligence could be further refined by employing advanced sentiment analysis models or even transformer-based models like BERT or GPT that understand deeper nuances of human language. This would enable more accurate emotional detection and allow for more personalized responses.

The content base can also be expanded by including more stories from Indian mythology and other scriptures, offering a wider variety of moral themes and teachings. Moreover, creating a user profile system where the chatbot remembers past conversations could provide continuity and a more personalized spiritual journey for returning users.

Another area of development is the addition of gamification elements, such as quizzes, interactive story choices, or badges for engaging with the content. These features could increase user retention and encourage deeper exploration of the teachings.

Lastly, this platform could be extended into a mobile app or deployed in regional languages, making it accessible to a broader audience across India and beyond. Multilingual support would particularly enhance its cultural relevance and inclusivity.

In conclusion, this project lays the groundwork for a unique blend of AI-driven interaction and traditional wisdom, and with future advancements, it has the potential to evolve into a powerful educational, emotional, and spiritual tool.

Chapter 7

References

[1]S. M. Metev and V. P. Veiko, Laser Assisted Microtechnology, 2nd ed., R. M. Osgood, Jr., Ed. Berlin, Germany: Springer-Verlag, 1998.

Breckling, Ed., The Analysis of Directional Time Series: Applications to Wind Speed and Direction, ser. Lecture Notes in Statistics. Berlin, Germany: Springer, 1989, vol. 61.

S. Zhang, C. Zhu, J. K. O. Sin, and P. K. T. Mok, "A novel ultrathin elevated channel low-temperature poly-Si TFT," IEEE Electron Device Lett., vol. 20, pp. 569–571, Nov. 1999.

M. Wegmuller, J. P. von der Weid, P. Oberson, and N. Gisin, "High resolution fiber distributed measurements with coherent OFDR," in Proc. ECOC'00, 2000, paper 11.3.4, p. 109.

R. E. Sorace, V. S. Reinhardt, and S. A. Vaughn, "High-speed digital-to-RF converter," U.S. Patent 5 668 842, Sept. 16, 1997.

(2002) The IEEE website. [Online]. Available: http://www.ieee.org/

[7] M. Shell. (2002) IEEEtran homepage on CTAN. [Online]. Available: http://www.ctan.org/tex-archive/macros/latex/contrib/supported/IEEEtran/

INDIVIDUAL CONTRIBUTION REPORT

“VIKRAM AND BETAAL”

Prajnabrata Mohanty
2105049

Abstract: This project presents an interactive web platform that harmoniously blends ancient Indian folklore with modern emotional well-being. Developed using HTML, CSS, JavaScript, and Python, the site is themed around the timeless tales of Vikram and Betaal, offering users a visually immersive and culturally rich storytelling experience. At its core is a thoughtfully designed chatbot that engages with users to understand their emotional state. In response, it suggests a fitting Bhagavad Gita verse, offering spiritual insight and comfort. Simultaneously, it presents a Vikram-Betaal story whose moral parallels the chosen verse, creating a unique bridge between mythology and meaningful introspection.
This fusion of narrative, spirituality, and technology aims to foster both cultural appreciation and personal reflection.

Individual contribution and findings: My individual contribution to this project involved designing and developing the "About Us" and "Contact Us" pages using HTML and CSS, with a focus on implementing a modern glass morphism design. This approach gave the pages a clean, translucent aesthetic that aligns with current UI trends, enhancing the visual appeal and user experience of the website. I ensured that both pages were fully responsive, accessible across devices, and stylistically consistent with the overall theme of the project. Through this process, I gained hands-on experience in advanced CSS techniques, particularly in layering, backdrop filters, and layout responsiveness.

Individual contribution to project report preparation: My contribution to the project report preparation mainly focused on creating the report by getting info on the parts I did on the project and acquiring the rest about the working by respective team members.

Individual contribution for project presentation and demonstration: For this part, I was responsible for creating the "About Us" and "Contact Us" pages using HTML and CSS, with a focus on design for a clean and modern interface. I ensured both pages were responsive and visually aligned with the theme of the site.
During the presentation, I demonstrated the design elements I implemented and explained how glass morphism was achieved using CSS techniques like backdrop-filter and transparency.
This experience improved my front-end development skills and taught me the importance of design consistency and clear communication.

Full Signature of Supervisor   Full Signature of Student

INDIVIDUAL CONTRIBUTION REPORT

“VIKRAM AND BETAAL”

Anurag Panda
2105181

Abstract: This project introduces an engaging web platform that thoughtfully merges the essence of ancient Indian mythology with contemporary emotional wellness. Built using HTML, CSS, JavaScript, and Python, the website is inspired by the legendary tales of Vikram and Betaal, delivering a visually appealing and culturally immersive storytelling environment. At the heart of the experience lies an intelligent chatbot designed to interact with users, gauging their emotional state through natural conversation. Based on this input, the chatbot offers a relevant verse from the Bhagavad Gita to provide spiritual perspective and comfort. Alongside this, it shares a Vikram-Betaal story that echoes the moral of the selected verse, creating a powerful link between timeless wisdom and personal reflection.
By weaving together narrative tradition, spiritual guidance, and modern technology, the platform encourages users to explore cultural heritage while gaining insight into their own emotional journeys.

Individual contribution and findings: My individual contribution to this project included the development of the "Home Page" and "Comics Page," along with its backend functionality. I focused on creating an intuitive and visually engaging user interface using HTML, CSS, and JavaScript, ensuring responsiveness and consistency with the overall theme. On the backend side of the Comics Page, I implemented features to support dynamic PDF viewing and integrated comment functionality using FastAPI. This gave users a seamless experience across different sections of the site. Throughout the process, I gained practical experience in frontend-backend integration, REST API development, and creating responsive layouts aligned with modern web design practices.

Individual contribution to project report preparation: My role in the project report preparation involved documenting my work on the Home Page, Comics Page, and backend integration. I collaborated with team members to collect their contributions and ensured that all technical components I worked on were clearly explained and well-structured in the report.

Individual contribution for project presentation and demonstration: For this part, I presented the "Home Page" and the "Comics Page" along with its backend flow. I explained how different components were integrated to deliver a cohesive experience, including the PDF viewer, comment submission feature, and overall navigation. I also demonstrated the complete website integration, ensuring all parts functioned together smoothly. This experience strengthened my understanding of full-stack development and improved my ability to communicate technical ideas effectively.

Full Signature of Supervisor　　　　　　　　　　Full Signature of Studen

INDIVIDUAL CONTRIBUTION REPORT

“VIKRAM AND BETAAL”

Ankit Hati
21052897

Abstract: This project presents a dynamic web platform that artfully blends the richness of ancient Indian mythology with the principles of modern emotional well-being. Developed using HTML, CSS, JavaScript, and Python, the site draws inspiration from the iconic stories of Vikram and Betaal, offering users a captivating and culturally immersive storytelling experience. Central to the platform is an intelligent chatbot that engages users in meaningful dialogue to understand their current emotional state. Based on this interaction, the chatbot recommends a thoughtful verse from the Bhagavad Gita, aimed at offering spiritual support and clarity. It further complements this with a Vikram-Betaal tale that aligns with the moral of the selected verse, creating a seamless narrative bridge between spiritual teachings and reflective introspection.
Through the fusion of traditional storytelling, philosophical depth, and cutting-edge technology, the project not only preserves cultural legacy but also promotes emotional insight and personal growth.

Individual contribution and findings: My individual contribution to this project involved developing both the front-end and back-end of the chatbot feature. On the front-end, I focused on creating an interactive and user-friendly interface that allowed users to communicate smoothly with the chatbot. On the back-end, I implemented the logic using Python and integrated it with the front-end via API calls to enable real-time interactions. Additionally, I was responsible for researching and collecting relevant data required for the chatbot’s responses. This data was cleaned, organized, and transformed into a structured data frame to be used efficiently within the system. Through this, I gained valuable experience in data handling, full-stack development, and building intelligent user-interactive components.

Individual contribution to project report preparation: My contribution to the project report preparation included documenting the work I carried out on the chatbot's front-end and back-end, as well as detailing the data research and data frame creation process. I ensured my sections were clearly articulated and collaborated with the rest of the team to incorporate their work into the report effectively.

Individual contribution for project presentation and demonstration: For the project presentation, I demonstrated the working of the chatbot, explaining both the design of the user interface and the backend logic powering the responses. I also discussed the process of gathering and structuring the data that fuels the chatbot’s accuracy and relevance. This experience enhanced my skills in explaining technical implementations clearly and highlighted the importance of combining effective UI with solid data-driven backend systems.

Full Signature of Supervisor   Full Signature of Student

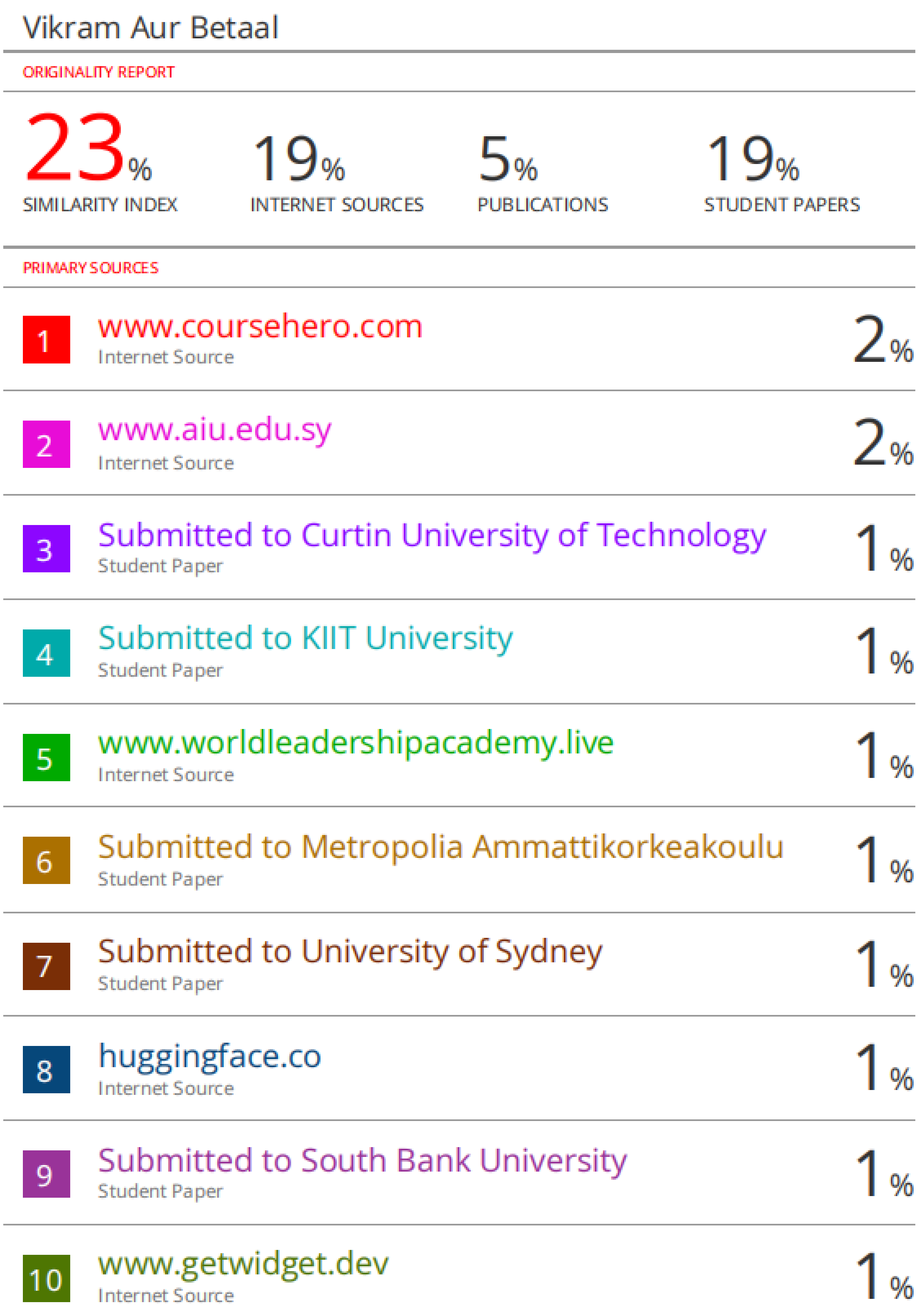
CONTRIBUTION

ANURAG PANDA (2105181):
1. Home Page
2. Comics Page
3. Integration of complete website.

ANKIT HATI (21052897):
1. Chatbot Frontend & Backend.
2. Data Research and Data frame Creation.

PRAJNABRATA MOHANTY (2105049):
1. About Us Page.
2. Contact Us Page.
3. Front-end and Back-end for both.

PLAGIARISM REPORT



Vikram Aur Betaal

ORIGINALITY REPORT

23%	19%	5%	19%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.coursehero.com Internet Source	2%
2	www.aiu.edu.sy Internet Source	2%
3	Submitted to Curtin University of Technology Student Paper	1%
4	Submitted to KIIT University Student Paper	1%
5	www.worldleadershipacademy.live Internet Source	1%
6	Submitted to Metropolia Ammattikorkeakoulu Student Paper	1%
7	Submitted to University of Sydney Student Paper	1%
8	huggingface.co Internet Source	1%
9	Submitted to South Bank University Student Paper	1%
10	www.getwidget.dev Internet Source	1%

13	medium.com Internet Source	1%
14	Submitted to Universiti Tunku Abdul Rahman Student Paper	1%
15	designrfix.com Internet Source	1%
16	www.templatecommunity.com Internet Source	1%
17	Submitted to University of Stirling Student Paper	1%
18	Submitted to National College of Ireland Student Paper	1%
19	itronixsolutions.com Internet Source	1%
20	usefil.eu Internet Source	1%
21	www.shura.org.tr Internet Source	1%
22	potatopirates.game Internet Source	<1%
23	Submitted to October University for Modern Sciences and Arts (MSA) Student Paper	<1%
24	Submitted to Northern Arizona University Student Paper	<1%
25	Submitted to University of Greenwich Student Paper	<1%
26	linux.feeder.ww7.be Internet Source	<1%
27	university.webflow.com Internet Source	<1%