



Dream Blog

Online Story Writing and Sharing Website

Hoang Nhat Duy Nguyen

BACHELOR'S THESIS
April 2023

Degree Programme in Software Engineering

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Degree Programme in Software Engineering

NGUYEN, HOANG NHAT DUY
Dream Blog
Online Story Writing and Sharing Website

Bachelor's thesis 38 pages, of which appendices 5 pages
April 2023

This thesis described the establishment of a website named 'Dream Blog' for writing and sharing dreams-related experiences. This project intended to inspire users to create their own dream-related stories by drawing inspiration from the fascinating and enigmatic world of dreams. In addition to showcasing the user interfaces of the website, the thesis also introduced the development environment, frameworks, and deployment services that were utilized to create it.

The React.js framework was the major environment used for platform development, and Visual Studio Code was used as the platform for coding and maintenance. A user-friendly and delightful experience was ensured by using the renowned front-end framework Bootstrap when designing the website and user interface. In order to manage the storage of user-generated stories and display those on the application front end, JSON Server was used as the database management service. The 'Dream Blog' front end and back end were hosted on Netlify and Glitch, respectively, to make the project accessible to users. These hosting services were selected due to their dependability and safe solutions for publishing and deploying applications.

The project provided straightforward user interfaces and all the necessary functions, achieving the goals outlined at the start of the project. It was distributed to numerous users, which led to the addition of stories to the platform. To give a thorough introduction to the development environment, frameworks, and services addressed in this thesis, a literature review of a variety of papers, theses, studies, and websites was conducted. The application's initial version was finished successfully, however there was still room for modifications and enhancements. Thus, plans were made to further develop the application in the near future.

Key words: front-end website application, react, bootstrap, json-server, front-end and back-end deployment services.

CONTENTS

1	INTRODUCTION.....	5
2	INSPIRATION, BACKGROUND AND TARGETS.....	6
	2.1 Project Inspiration.....	6
	2.2 Project Background and Objectives.....	7
3	DESIGNS.....	10
	3.1 User Interface.....	10
	3.2 Home Page.....	10
	3.3 Stories Zone and Stories Detail.....	12
	3.4 Create New Stories.....	14
4	DEVELOPING ENVIRONMENT, FRAMEWORK AND SERVICES.....	17
	4.1 Visual Studio Code.....	17
	4.2 React.js.....	19
	4.3 Bootstrap.....	20
	4.4 JSON and JSON Server.....	23
	4.5 Deployment Services.....	25
	4.5.1 Netlify.....	25
	4.5.2 Glitch.....	29
5	DISCUSSION.....	31
	APPENDIX.....	34
	Appendix 1. The use of React in coding and developing.....	34

ABBREVIATIONS AND TERMS

API	Application Programming Interface
CDN	Content Delivery Network
CSS	Cascading Style Sheets
db	Database
DOM	Document Object Model
GIT	Global Information Tracker
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
JSON	JavaScript Object Notation
MVC	Model-view-controller
OS	Operating System
PHP	Hypertext Preprocessor
REM	Rapid Eye Movement
REST	Representational State Transfer
SPA	Single Page Application
TAMK	Tampere University of Applied Sciences
UI	User Interface
URL	Uniform Resource Locator

1 INTRODUCTION

Long viewed as a window into the unconscious mind, dreams can reveal our suppressed hopes, concerns, and experiences. In spite of this popular misconception, research into the science of dreaming is still in its infancy, and many uncertainties remain regarding the nature and purpose of the dream experience.

Dream-sharing with others may be a private and occasionally uneasy experience. While some people may choose to voluntarily discuss their aspirations with close friends or family, others may decide to do so on social networking sites like Facebook or Twitter, running the risk of criticism from others. A fascinating new initiative named 'Dream Blog' has been developed as a personal and thesis project in light of these complications, with the realm of dreams serving as its main emphasis.

The 'Dream Blog' project intends to offer a web space for publishing and exchanging tales based on individual dreaming encounters. This thesis presents a thorough examination of the design, development environment, frameworks, and deployment services used in constructing the application and explores the reasoning for choosing dreams as the project's topic. This thesis intends to contribute to the knowledge of dreams as an interesting topic and the development of the application by shining light on the project's logic and technological components.

2 INSPIRATION, BACKGROUND AND TARGETS

2.1 Project Inspiration

We frequently experience images, feelings, ideas, and other things when we sleep, all of which are categorized under the term 'dream'. Dreams can range from being extremely vague, fleeting, perplexing, or even boring to being extremely vivid or emotionally charged. While some dreams are joyful, others are spooky or downcast. There are times when dreams seem to have a clear storyline, even though many of them seem to make no sense at all (Kendra Cherry 2022).

There are several theories about the nature of dreams. These are the three famous theories:

- Psychoanalytic theory: In accordance with Sigmund Freud's theory of dreams, dreams might symbolize unconscious urges, wishes, and desires. According to Freud, people are motivated by repressed and unconscious longings, such as violent and sexual urges (Kendra Cherry 2022).
- Cognitive theory: This study asserts that dreams do in fact help to organize and make sense of waking life. It focuses on how the bulk of dreams seem to represent ordinary, everyday circumstances (Catherine A. Sanderson 2022).
- Activation-synthesis theory: Introduced in 1977 by J. Allan Hobson and Robert McCarley (Harvard psychiatrists), proposes that during rapid-eye-movement (REM) sleep, brain pathways become engaged, causing the amygdala and hippocampus to produce a variety of electrical impulses (Kendra Cherry 2022).

According to some of the more well-known dream theories, having dreams serves as a way to:

- Consolidation of memories: Dreams help us sort through complex ideas and feelings, store essential memories and lessons we have learned, eliminate unimportant memories, and store important memories (James Roland 2023). According to research (Björn Rasch & Jan Born 2013), sleep aids in memory

retention. We will be able to recall new information more easily if we learn it and then give it some time to sink in while sleeping.

- Processing feelings: We might use our dreams as a means to deal with emotional conflicts in our lives. Additionally, our brain may form connections about our feelings that our conscious self wouldn't make because it is functioning at much higher emotional level than when we are awake (Michael S. Franklin & Michael J. Zyphur 2005).
- Improve our creativity: One explanation for why we dream is that it stimulates our creative instincts (Michael S. Franklin & Michael J. Zyphur 2005). All types of artists acknowledge that some of their most imaginative work has been inspired by dreams. We could have occasionally woken up in our lives with a brilliant idea for a song or a movie.
- Get familiar with potential threats: The amygdala is one of the brain regions that is most active while dreaming. This brain region is responsible for the fight-or-flight response and the survival instinct. The amygdala is more active while we sleep than it is when we are up; therefore, it might be the brain's method of preparing us for a threat (James Roland 2023).

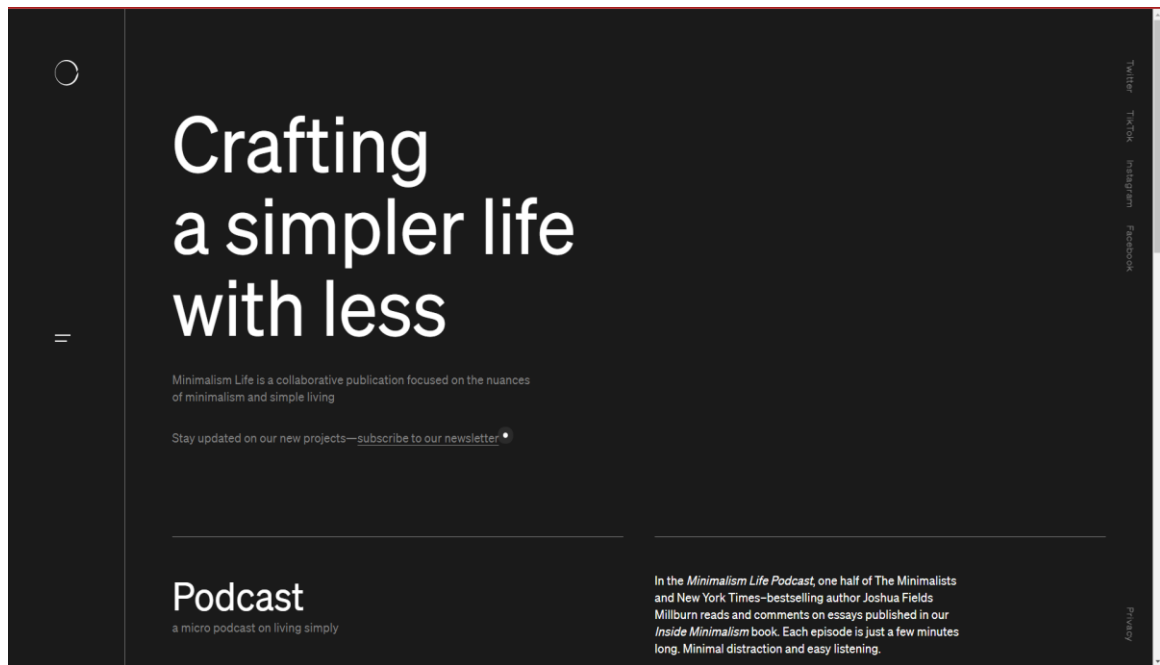
Overall, dreams are intricate occurrences that take place while we sleep and might include images, concepts, and emotions. Dreams are said to be used for a variety of things, including memory consolidation, emotional processing, creativity stimulation, and becoming familiar with potential dangers, according to theories like psychoanalytic, cognitive, and activation-synthesis.

2.2 Project Background and Objectives

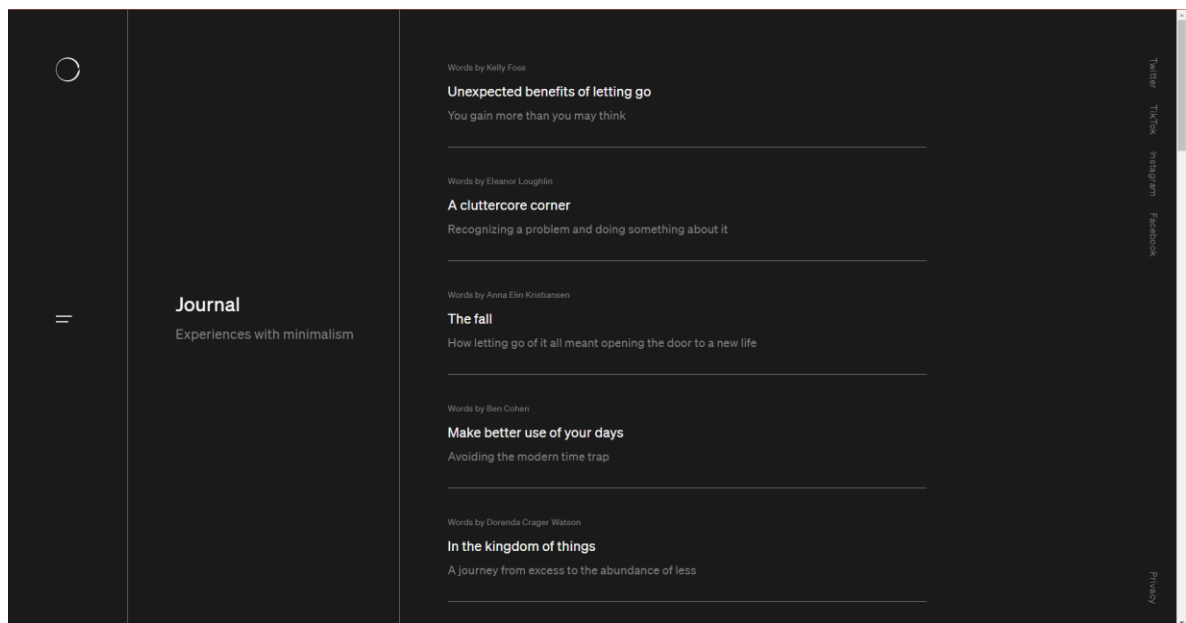
There are a wide variety of websites on the internet that offer articles and suggestions on the subject of dreams (<https://cmloegcmluin.wordpress.com/dream-blog/>, <https://wakefully.io/blog>, ...). However, there do not seem to be many forums where people can openly discuss and share their own unique dreaming experiences. In order to cover this gap, 'Dream Blog', an online platform created exclusively for writing and sharing stories based on individual dream encounters, was created.

The major goals of this website are twofold: first, to provide users with a website with a user-friendly interface, smooth user experiences, and the functionality they need, such as the ability to create and publish original stories based on their individual dreaming experiences; and second, to give users the choice to stay anonymous while sharing their stories. 'Dream Blog' aims to develop a community of dream enthusiasts who can interact and engage with one another through the fascinating realm of dreams by offering a secure and imaginative platform for people to express and exchange their various dreaming stories.

The popular website Minimalism Life served as an influence for the idea behind 'Dream Blog' (Picture 1). Users of Minimalism Life get access to a platform where they may read journals posted by website contributors, listen to podcasts, and read an eBook with 50 brief but thought-provoking essays. 'Dream Blog' was designed with the user-friendly interface and minimalist aesthetic of Minimalism Life in mind, with a similar seamless and simple experience in mind for users to share and explore their individual dream narratives. The goal of 'Dream Blog', which places the emphasis on the stories and experiences shared by users and makes it accessible and pleasurable for people to engage with the fascinating realm of dreams, was in line with the minimalist philosophy of Minimalism Life.



PICTURE 1. Minimalism Life website (<https://minimalism.com/>).



PICTURE 2. A list of shared stories written by Minimalism Life contributors (<https://minimalism.com/journal>).

3 DESIGNS

3.1 User Interface

The application, which caters to a broad range of users from young children to teenagers, adults, and beyond, has been created to be as user-friendly as possible. No matter their level of technical expertise, anyone can easily browse and use the application thanks to the design philosophy's emphasis on simplicity. In order to accomplish this, a minimalist design strategy is utilized, with a clear and uncluttered layout that places the content front and center.

The use of color is one of the design's main components. The application uses black, white, and orange gold (#D4A017) as its main colors to create a color scheme that is both eye-catching and simple to read. The bright, bold orange gold used for titles, headers, navigation bar texts, navigation bar buttons, background color for buttons, and card borders contrasts sharply with black, which is primarily used as a backdrop. This produces a visually stunning contrast that grabs attention and aids in directing users through the application. White is used for paragraph text and placeholders in comparison to the boldness of the black and orange gold, giving the page a clean and elegant appearance that is both simple to read and aesthetically pleasing.

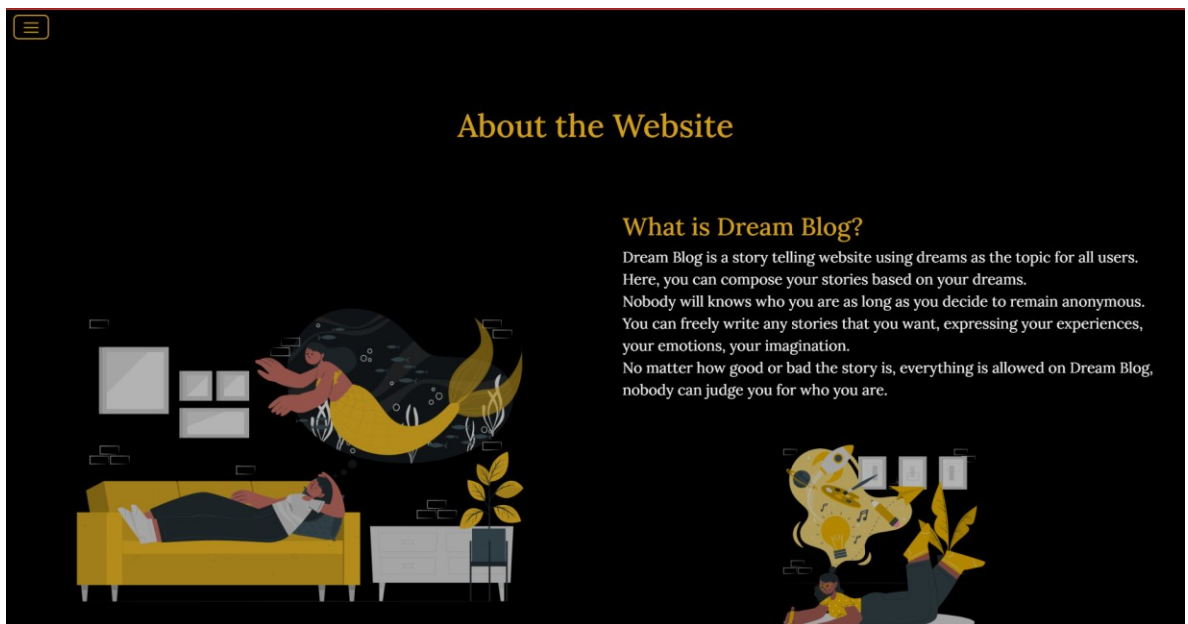
3.2 Home Page

Users will note that the application's home page is divided into three separate areas when they first go there. Each of these sections has a particular function and offers users useful information. The "Introduction" section is the first one, and it provides a brief summary of the application's goals (Picture 3). The next section is "About the Website," where users may read more about the purposes of the application, their goals, and the sources of their inspiration (Picture 4). The "Website Features" section, which is the last section, lists all the features and functionalities you can anticipate from the program and explains how they can help the users. This section

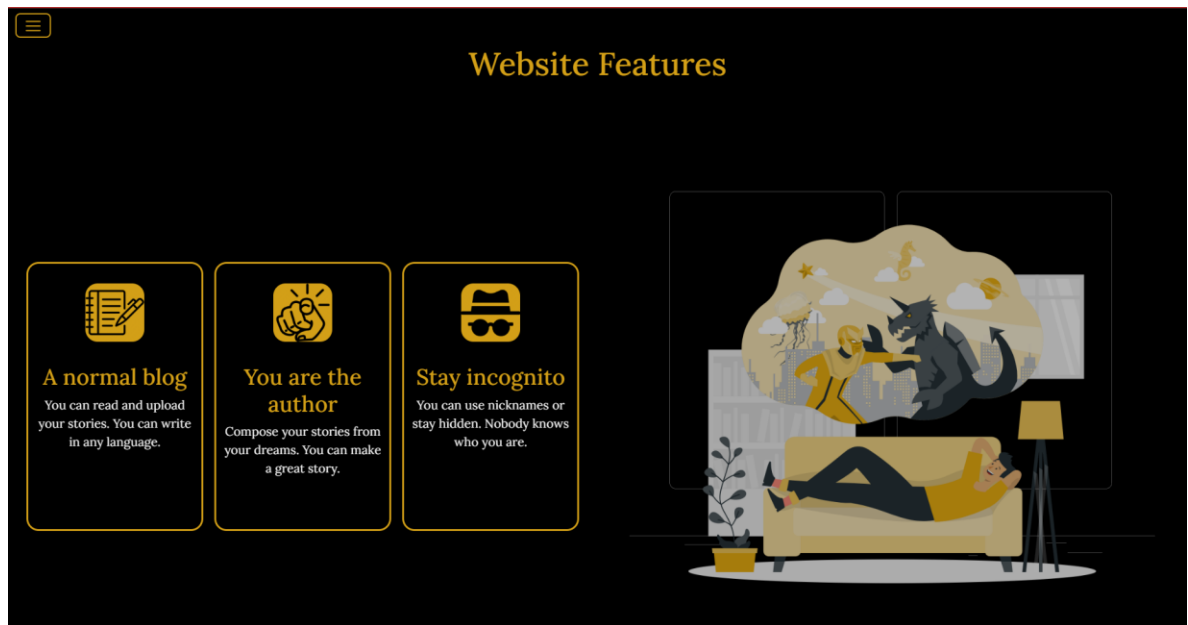
will also give them a better understanding of its capabilities and how it can help (Picture 5). All in all, all three elements function in unison to deliver a thorough and interesting user experience that will give users peace of mind.



PICTURE 3. Home page “Introduction” section.



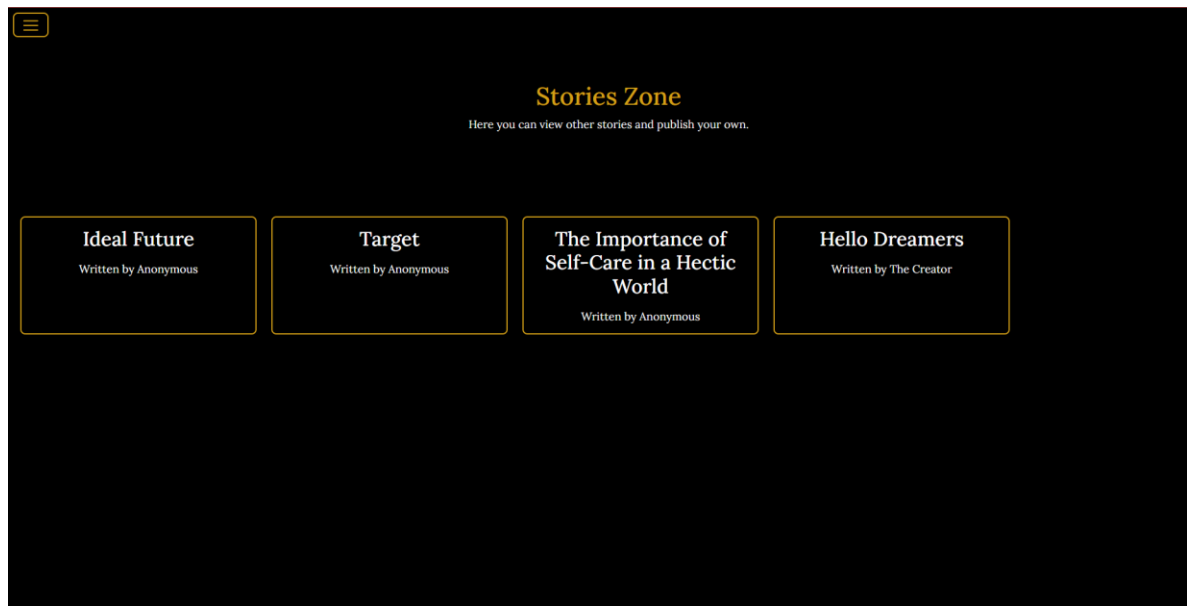
PICTURE 4. “About the Website” view on website Home page.



PICTURE 5. Home page “Website Features” section.

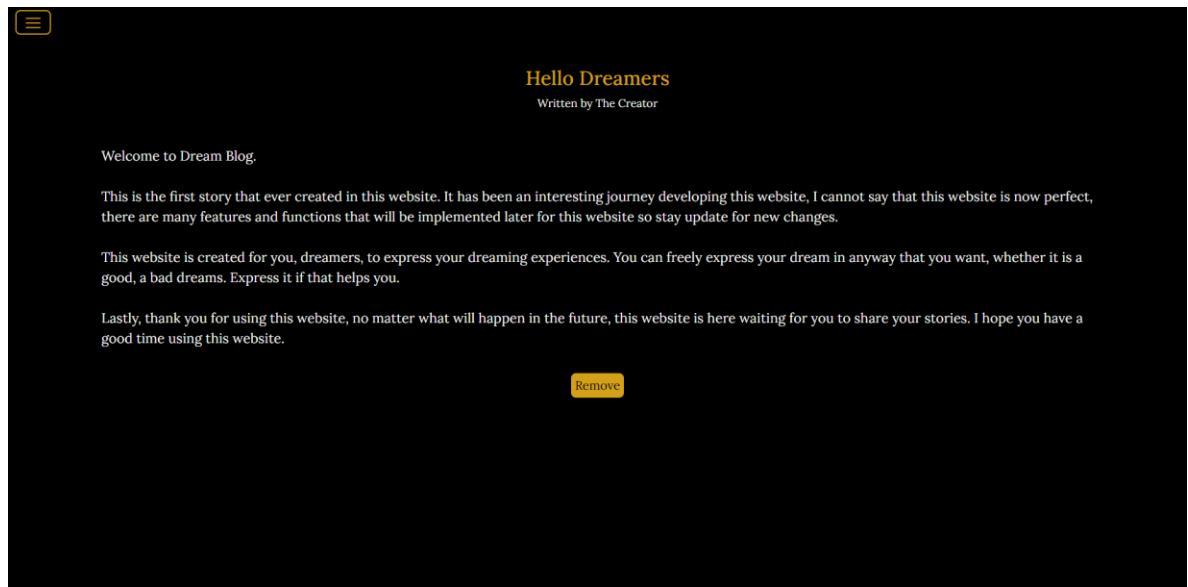
3.3 Stories Zone and Stories Detail

A special page dubbed the “Stories Zone” has been established on ‘Dream Blog’ to enable quick access to both user-shared stories and personal experiences. The title and author of user-written stories are displayed in preview cards on this page. According to the website's privacy policy, the author's default display is set to anonymous or nicknames in order to protect user privacy. Users can choose a nickname of their choice or remain anonymous when sharing their stories, giving them the opportunity to maintain the level of anonymity they prefer while interacting with the community (Picture 6).



PICTURE 6. "Stories Zone" page. Currently there has been four stories created by users.

When a story is chosen from the "Stories Zone," viewers are routed to a special website where, using the story ID, they can read the entire content of the narrative. The following "Stories Detail" page (Picture 7) demonstrates how users can now explore the story's specifics. In addition, users can currently remove their stories from the database by clicking the "Remove" option underneath the narrative content. It has been acknowledged that users may abuse this function to delete stories they are not interested in. Therefore, the button will soon be removed from the site as a precaution to guarantee data integrity and stop abuse.

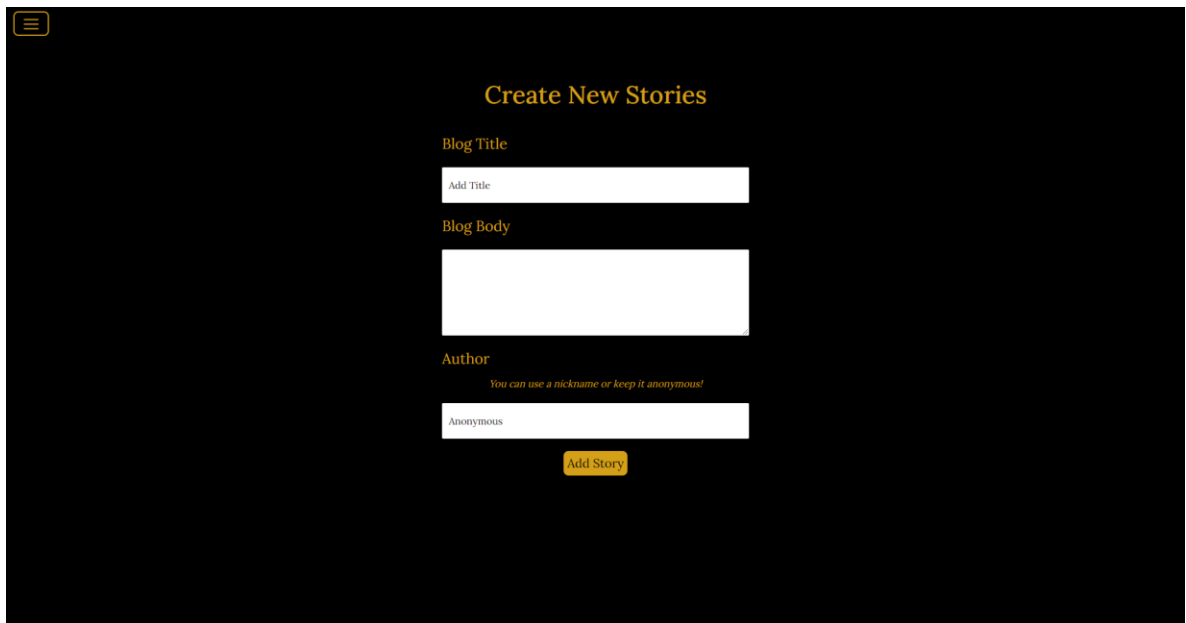


PICTURE 7. “Stories Detail” view. Story content can be viewed after users selected a story.

3.4 Create New Stories

The “Create New Stories” page was created with the intention of giving people a simple way to write and share their stories (Picture 8). The title, content, and author of the narrative are entered into text boxes on the page's straightforward form. In the author input field, users have the choice to utilize nicknames or maintain anonymity.

This simplified process guarantees that users may quickly write and submit their articles without encountering any unneeded difficulties. The “Create New Stories” page seeks to promote and enable the sharing of unique and interesting stories from users around the world by offering a user-friendly interface for story creation.



PICTURE 8. “Create New Stories” page, users can freely express their dreaming experiences and publish them as stories.

3.5 Future Development

Reviewing the present design of the website, it is evident that new features could be added to make it better. The ability for anyone to delete stories without the creator's permission is a concern; there is a chance that this problem will lead to the deletion of user-written stories that took a lot of time and effort to write. Therefore, a remedy that demands users register an account has been put in place to address this issue, giving users more control over the content they produce and upload to the website. This new feature will make sure that user-generated material is safeguarded and preserved by requiring authentication before stories can be removed.

The “Create New Stories” page is another place that needs focus. Users can currently only add a story's title and content, but this feature will soon be expanded to allow users to add other details like the story's creation date, a special story ID, and more. The website seeks to improve the user experience and make it simpler for users to navigate and search for the content they are interested in by offering more detailed information about each story.

An extra interesting feature that will be added to the application “Stories Detail” view is an “Edit Story” button that will allow users to edit the stories they have written on their own. Users can edit the text, add or remove photographs, and update the story's metadata by clicking this button in order to make the necessary adjustments to the content of their stories. Users will have more freedom and control over their content thanks to this new functionality, enabling them to develop and improve their stories over time. This option is made to give users the power to constantly expand and enhance their stories, guaranteeing that their imaginative work stays alive and compelling to readers.

Despite these changes, it is crucial to safeguard users' privacy and safety. As a result, even after users create accounts, the website will enable anonymous or nickname-based authorship. By doing this, users are given the peace of mind to tell their stories without worrying that their identity will be compromised.

4 DEVELOPING ENVIRONMENT, FRAMEWORK AND SERVICES

4.1 Visual Studio Code

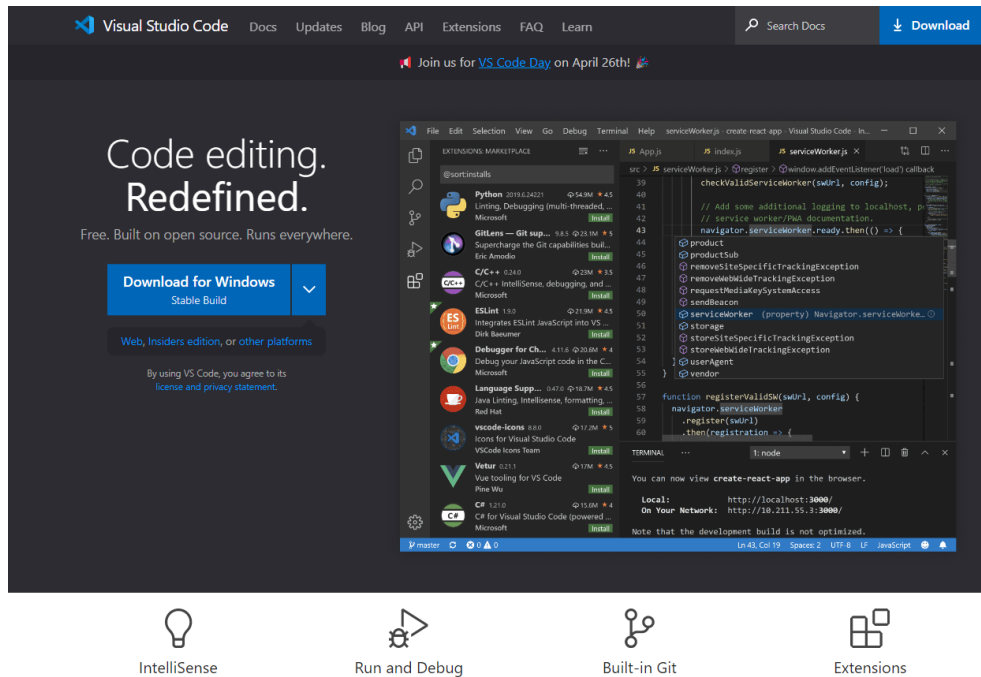
Visual Studio Code is a free and powerful source code editor that is widely used on local devices like desktops and browsers. It is suitable and frequently used for many operating systems (OS), for example, Windows, Mac, Linux, or Raspberry Pi. The editor supports JavaScript, TypeScript, and Node.js with built-in features. Programming languages, such as Java and Python; runtimes, such as .NET and Unity; environments, such as Docker and Kubernetes; or cloud services, such as Amazon Web Services and Microsoft Azure, are just a few of the extensions available for Visual Studio Code. (Martin Heller 2022).

Visual Studio Code provides built-in source code control, including Git support, linting, multi-cursor editing, parameter suggestions, and other potent editing capabilities, in addition to the basic idea of being lightweight and starting quickly. Additionally, it offers code completion using IntelliSense for variables, methods, and imported modules. A substantial portion of this was changed using Visual Studio technology. (Martin Heller 2022).

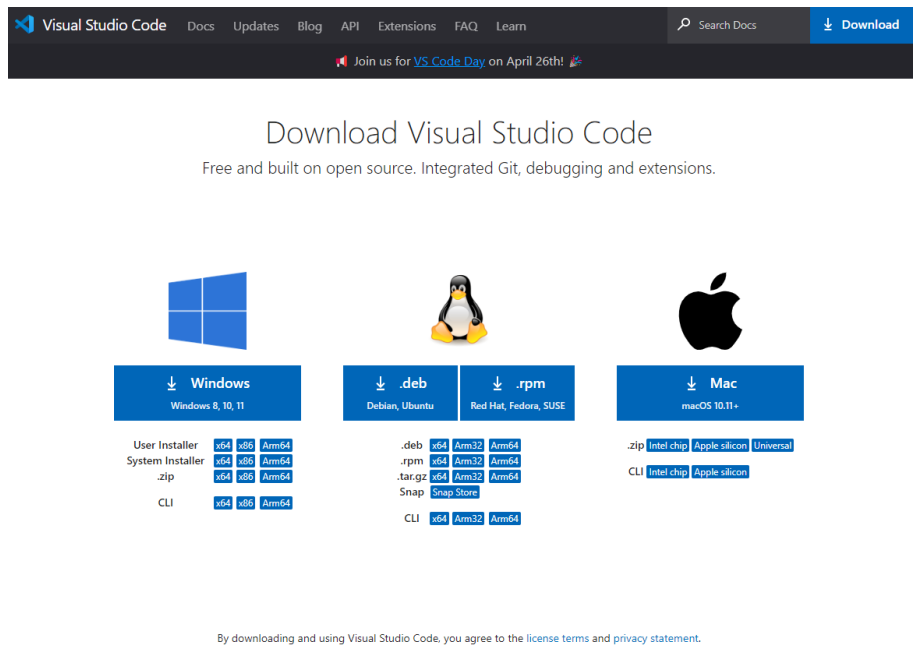
Visual Studio Code itself was created using Node.js, TypeScript, the Language Server Protocol, and the Electron shell. It is updated monthly. When necessary, the multiple extensions are updated. From simple syntax highlighting and bracket matching through debugging and refactoring, many programming languages and their extensions provide differing degrees of assistance. Users can add rudimentary support for their favorite language using TextMate colorizers if there is no language server available. (Martin Heller 2022).

Visual Studio Code can be downloaded from either the Visual Studio Code website home page (Picture 9) or, for greater control, its download page (Picture 10). Depending on your system, 'apt', 'apt-get', 'rpm', 'yum', or 'snap' are additional command-line tools that you can use to install software on Linux or Raspberry Pi

OS. There are installation guidelines for Windows, Linux, macOS, and Raspberry Pi. Additional tools might need to be installed, such as Git, Node.js, TypeScript, a C++ compiler, Python 3.7 or later, Yeoman, and/or some version of .NET, in order to support your source code manager and programming languages. (Martin Heller 2022).



PICTURE 9. Visual Studio Code website home page (<https://code.visualstudio.com/>).



PICTURE 10. Visual Studio Code website download page (<https://code.visualstudio.com/Download>).

4.2 React.js

Facebook created the open-source React JavaScript library. It is used to quickly and efficiently construct interactive user interfaces and online apps in comparison to utilizing pure JavaScript. With React, developers create applications by creating reusable components. When combined, these pieces, which are individual components of a final interface, create the application's whole user interface. (David Herbert 2022).

React manages the view layer of an application by providing the best and most efficient rendering execution, much like the V in a model-view-controller (MVC) paradigm. Instead of treating the entire UI as a single object, React.js encourages developers to deconstruct these complex user interfaces (UI) into separate, reusable components that act as the UI building blocks. By combining JavaScript speed and efficiency with a more efficient method of manipulating the Document Object Model (DOM), the React.js framework is able to generate web pages more fast and create highly dynamic and responsive online applications. (David Herbert 2022).

Developers can create what is referred to as a single-page application (SPA) with React. On the first request, a single-page application loads just one Hypertext Markup Language (HTML) file. Then, it uses JavaScript to change the necessary section, content, or body of the webpage. Because the client does not have to reload the entire webpage to receive a new page whenever a user submits a new request, this pattern is known as client-side routing. Instead of forcing a full page reload, React intercepts the request and just fetches and modifies the relevant areas. Better performance and a more dynamic user experience are the outcomes of this strategy (David Herbert 2022).

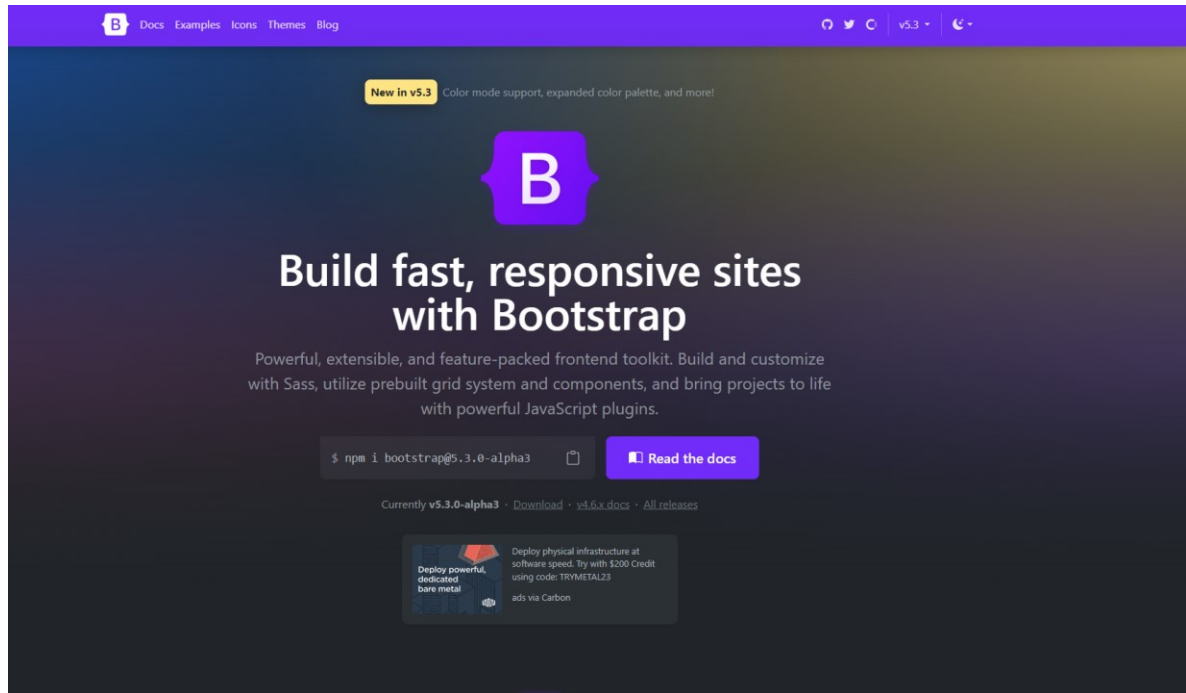
4.3 Bootstrap

Bootstrap (Picture 11) is a front-end programming and styling framework for building free and open-source websites and web applications. To facilitate the creation of responsive, mobile-first websites, it provides a collection of material for template designs. (Jordana A. 2022). With Bootstrap, developers can build any application in minutes, from a prototype to a finished product.

Bootstrap contains the fundamentals of responsive web development as a framework, so developers only need to insert code into a pre-defined grid system. Because of its ease of use, Bootstrap enables web developers to create websites much more quickly without having to spend time worrying about fundamental commands and functions. It consists of scripts built on the Hypertext Markup Language (HTML), cascading style sheets (CSS), and JavaScript platforms for various web design-related features and operations (Andrew Zola 2022).

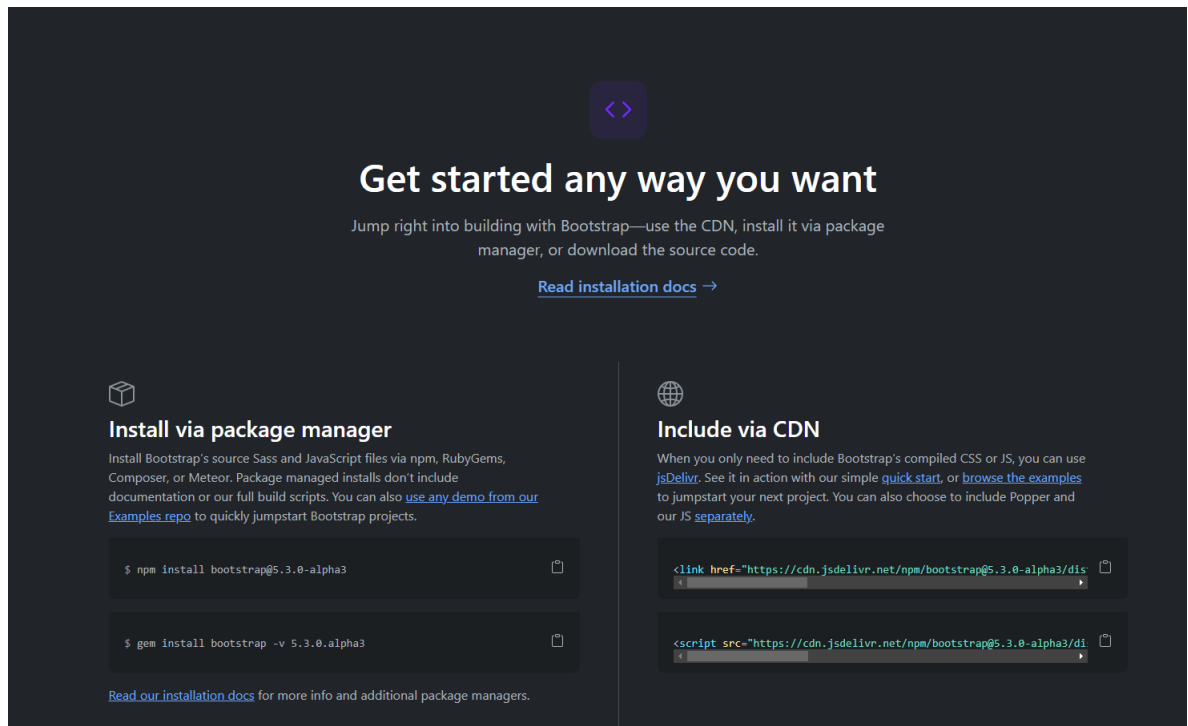
With the help of Bootstrap, it is less complicated for developers to work with their applications responsiveness across multiple devices, such as mobile phones, tablets, or small computer. Whenever a users enter an application on a different screen, Bootstrap automatically adjust the application display based on the screen's size and orientation. This can be understood as the mobile-first strategy, since it is

believed that most employees will primarily use different devices or applications to complete their tasks. Bootstrap also provides UI elements, layouts, JavaScript tools, and an implementation framework to meet the design requirements of diverse technologies. (Andrew Zola, 2022). Bootstrap comes in both precompiled and source code-based versions. The latter is preferred by seasoned developers since it enables them to modify the styles to meet their projects (Jordana A. 2022).

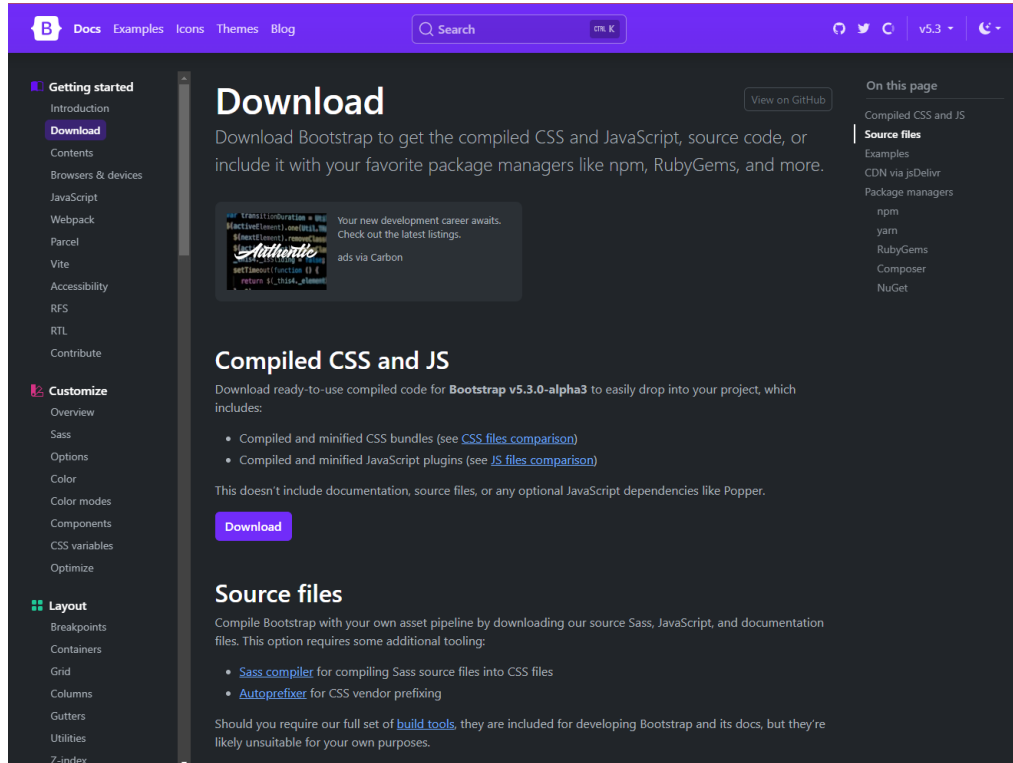


PICTURE 11. Bootstrap website home page, the current version of Bootstrap is version 5.3 (<https://getbootstrap.com/>).

Bootstrap can be installed via 2 methods (Picture 12): using a package manager with command lines such as “npm install bootstrap@5.3.0-alpha3” or “gem install bootstrap –v 5.3.0.alpha3”, or using a Content Delivery Network (CDN) when Bootstrap generated CSS or JavaScript is all that needs to be included. Developers can find more information and a guide about how to download Bootstrap for projects on the official Bootstrap website (Picture 13).



PICTURE 12. Installation methods provided by Bootstrap website. Developers can install via either package manager (left) or CDN (right) (<https://getbootstrap.com/>).



PICTURE 13. Download page where users can find more guide of how to use and install Bootstrap (<https://getbootstrap.com/docs/5.3/getting-started/download/>).

Developers are using Bootstrap more due to the benefits of utilizing it over other web development frameworks (TailWind CSS, Foundation, ...). Some of Bootstrap benefits are listed below:

- **Simple to use:** Bootstrap is easy to learn. Due to its popularity, there are a lot of tutorials and forums online that can help new developers get started. One of the reasons for Bootstrap's enormous appeal with web designers and developers is its simple file structure. Its files are arranged for easy access, and all that is required to alter them is a working understanding of HTML, CSS, and JavaScript. (Jordana A. 2022).
- **Flexible Grid:** Bootstrap comes with a pre-built grid system, saving users the time and effort of having to build one from scratch. Media queries should not be included in the CSS file. With the help of the grid system, which consists of rows and columns, programmers can add a grid on top of an existing one. The grid system in Bootstrap simplifies the data entry method. It has a variety of media queries, allowing users to specify the unique breakpoints for each column in accordance with the requirements of their web project (Jordana A. 2022).
- **Browser Compatibility:** The bounce rate can be decreased, and developer's websites can rank higher in search results by being accessible via multiple browsers. Bootstrap satisfies that criteria by working with the most recent iterations of widely used browsers (such as Google Chrome, Microsoft Edge, ...). Websites using Bootstrap should work properly on lesser-known browsers like WebKit and Gecko despite not supporting them, but there might be restrictions (Jordana A. 2022).

4.4 JSON and JSON Server

JavaScript Object Notation (JSON) is a free, open-standard file format for data exchange. It transfers and stores data items using text that can be read by humans. JSON consists of two readable attributes, for example, value pairs and arrays (Javatpoint n.d.). JSON is a widely ubiquitous data format with different applications.

Developers can quickly and easily develop sample Representational State Transfer (REST) JSON services using the Node Module JSON Server; all they need is some sample data in a JSON file (Javatpoint n.d.).

When developing front-end applications that must communicate with a server but lack a backend API, JSON Server is helpful. The creation of a fake REST API for usage during development and eventual replacement with a genuine API is simple with JSON Server. Additionally, without having to write any server-side code, JSON Server enables developers to quickly prototype and test new API endpoints. JSON Server can be installed through the system terminal with the command: “npm install –g json-server”. More installation guides and instruction on how to make a personal JSON-Server can be found at: <https://www.npmjs.com/package/json-server>.

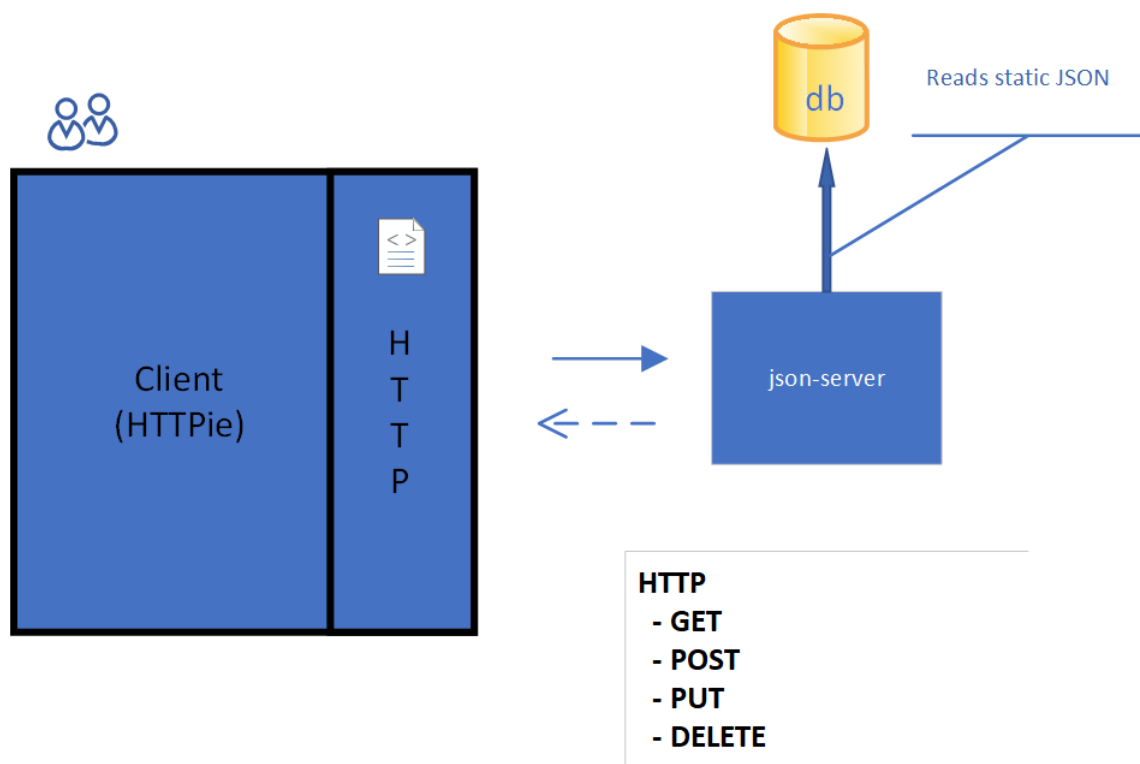


FIGURE 1. JSON-Server architecture illustration displaying the interaction between clients with the server (<https://dzone.com/articles/zero-code-rest-with-json-server>).

As seen in Figure 1, when a client (such as a web browser or mobile application) sends an HTTP request (POST, GET, PUT, or DELETE) to the JSON-Server, the server reads static JSON responses from a database (db) at startup. The returned

values are presented in JSON format on the client after the server has finished reading them. Due to the pre-loaded JSON data in the database, this enables clients to communicate with the server and receive responses.

4.5 Deployment Services

The front end and back end of the application are built using two deployment services, Netlify and Glitch, respectively, to ensure that 'Dream Blog' is a website that can be readily accessed and used by everyone. These services offer the infrastructure and resources required to develop and host the website, enabling users to access it from a variety of devices and places.

4.5.1 Netlify

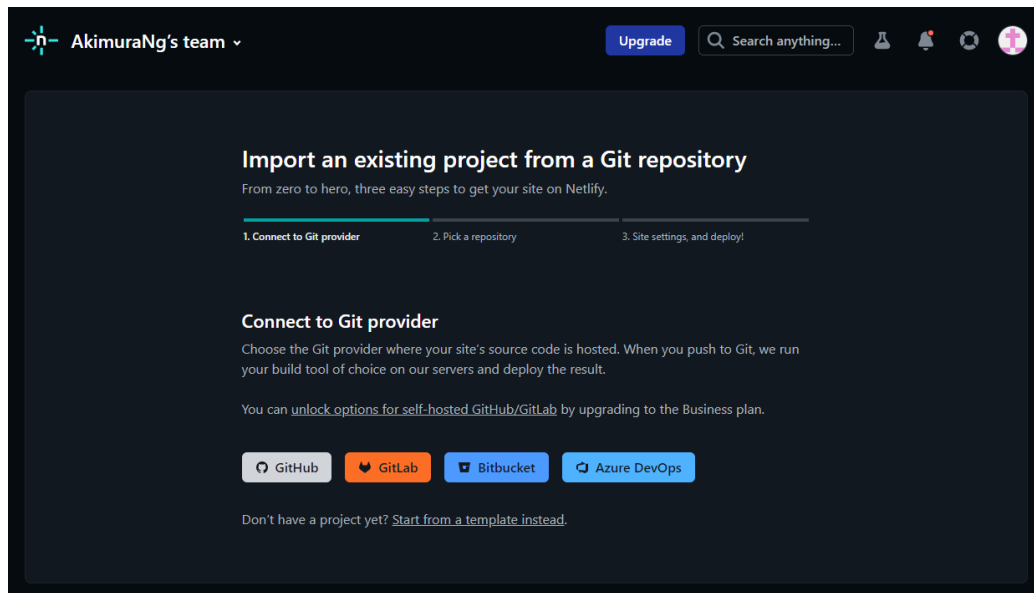
The San Francisco-based web hosting infrastructure and automation technology provider Netlify offers very cost-effective web hosting and automation. Netlify makes the process of installing and hosting a website easier for programmers, as they would not have to spend a lot of time or effort on the job because it was done for them. Not only that, but Netlify also offers developers a number of advantages (Joel Varty 2020).

The fact that Netlify chooses the best CDN and distributes content is one of its best features. As a result, pre-built websites load quicker than websites hosted through traditional hosting networks. Instead of needing to load the website every time they visit, the user gets a pre-loaded copy of it straight from the closest server. This considerably reduces load times (Joel Varty 2020).

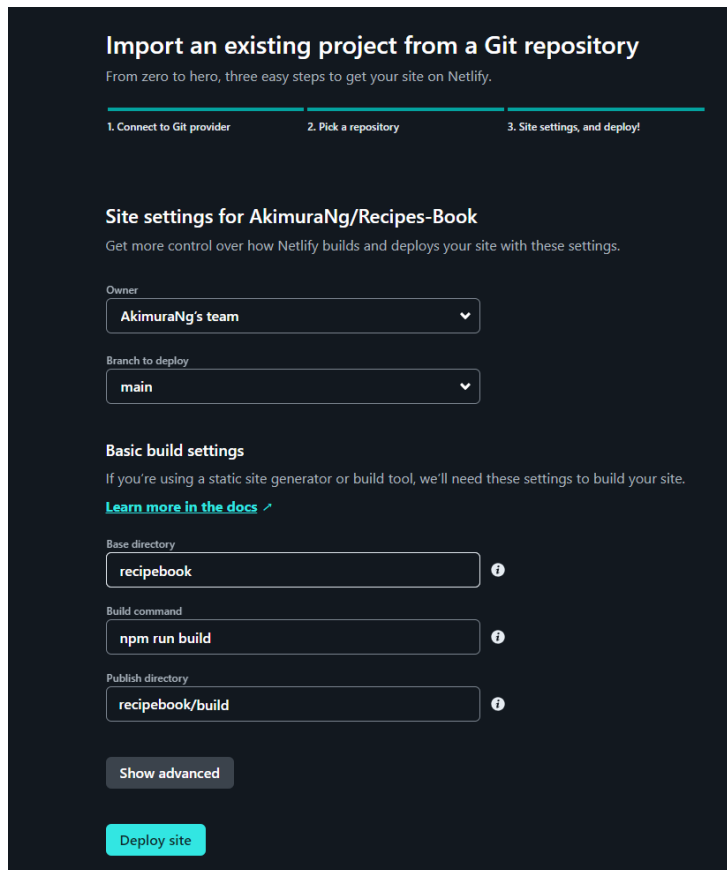
All of the pages are pre-rendered in static HTML by a build procedure that Netlify executes through the developer Git provider (GitHub, GitLab, ...) repository. Netlify builds its own microservices and a repository that uploads to Git provider. For the

purpose of delivering pre-built static websites to users, it then executes material and distributes it across a large CDN (Joel Varty 2020).

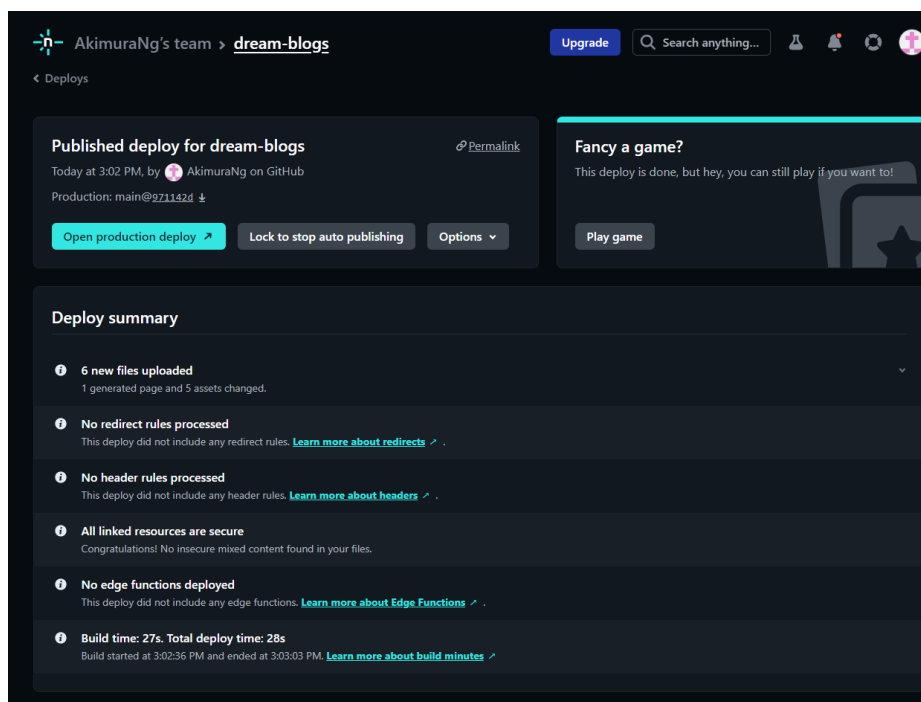
Simply connect Netlify to a repository, and it will automatically build the site (Picture 15), launch plugins, and publish each time a developer submits a commit. There is also a deploy summary (Picture 16) and a deploy log (Picture 17) where users can easily track the build process of their website. Netlify offers users the ability to make changes to the project site or create a custom domain name (Picture 18).



PICTURE 14. Users have the options to import their project from Git providers such as GitHub, GitLab, ... or start from a template.



PICTURE 15. After choosing a repository, Netlify will ask for the deployed branch and base directory, the build command will be automatically generated.



PICTURE 16. A summary of project deployment is showed for users to check.

```

Deploy log
Preview
48 3:02:55 PM: "create-react-app" at version "5.0.1"
49 3:02:55 PM: Section completed: initializing
50 3:02:57 PM:
51 3:02:57 PM: Netlify Build
52 3:02:57 PM:
53 3:02:57 PM:
54 3:02:57 PM: > Version
55 3:02:57 PM: @netlify/build 29.9.2
56 3:02:57 PM:
57 3:02:57 PM: > Flags
58 3:02:57 PM: baseRelDir: true
59 3:02:57 PM: buildId: 643a9258b803a20008438022
60 3:02:57 PM: deployId: 643a9258b803a20008438024
61 3:02:57 PM:
62 3:02:57 PM: > Current directory
63 3:02:57 PM: /opt/build/repo/dream
64 3:02:57 PM:
65 3:02:57 PM: > Config file
66 3:02:57 PM: No config file was defined: using default values.
67 3:02:57 PM:
68 3:02:57 PM: > Context
69 3:02:57 PM: production
70 3:02:57 PM:
71 3:02:57 PM: 1. Build command from Netlify app
72 3:02:57 PM:
73 3:02:57 PM:
74 3:02:57 PM: $ npm run build
75 3:02:57 PM: > dream@0.1.0 build
76 3:02:57 PM: > react-scripts build
77 3:02:58 PM: Creating an optimized production build...
78 3:02:58 PM: Browserslist: caniuse-lite is outdated. Please run:
79 3:02:58 PM: npx update-browserslist-db@latest
80 3:02:58 PM: Why you should do it regularly: https://github.com/browserslist/update-db#readme
81 3:03:02 PM: Browserslist: caniuse-lite is outdated. Please run:
82 3:03:02 PM: npx update-browserslist-db@latest
83 3:03:02 PM: Why you should do it regularly: https://github.com/browserslist/update-db#readme
84 3:03:03 PM: Compiled successfully.
85 3:03:03 PM:

```

PICTURE 17. Deploy log shows the build and deploy process of the project with time stamp and build status.

The screenshot shows the Netlify dashboard for the 'dream-blogs' project. At the top, there's a navigation bar with 'Upgrade' and a search bar. Below that, there's a 'Site overview' section with tabs for 'Deploys', 'Functions', 'Edge Functions', 'Integrations', 'Forms', 'Large Media', 'Split Testing', 'Analytics', 'Graph', and 'Site settings'. The main content area is divided into several sections:

- dream-blogs**: Shows the site URL 'https://dream-blogs.netlify.app', 'Deploys from GitHub', and 'Last published at 3:03 PM'. There are buttons for 'Site settings', 'Domain settings', and 'Unfavorite site'.
- Set up your site**: A section with three numbered steps:
 - 1 Your site is deployed ✓**: 'Try a test build and deploy, directly from your Git repository or a folder.'
 - 2 Set up a custom domain**: 'Buy a new domain or set up a domain you already own.'
 - 3 Secure your site with HTTPS**: 'Your site is secured automatically with a Let's Encrypt certificate.'
- With Netlify CLI, you can share your development server**: A tip that says 'over HTTPS'.
- Production deploys**: A list of deploys, with the most recent one being 'Production: main@971142d Published' at '3:02 PM: Fix input field, add navbar function'.
- Deployment Previews**: A list of preview deploys, with the most recent one being 'Deployment Preview #29: finalizing@f78787a' on 'Apr 13: Fix mobile.css'.

PICTURE 18. Site setting for 'Dream Blog' project, users can change the site name here or set up custom domain for the site.

4.5.2 Glitch

Similar to Netlify and other deployment services, Glitch is also a project building and hosting platform where users are allowed to create the website of their dreams. Glitch was introduced years ago, and until now it has attracted over a million contributors, developers, professionals, ... to participate in the Glitch community, developing incredible websites, bots, applications, artworks, virtual reality experiences, interactive infographics, and many other things (Anil Dash 2018). Developers may go to Glitch and create an entire project there, whether it be a static site using simply HTML and CSS, a front-end project using Angular and React, or a full-stack web using NodeJS (Beribey 2020).

Glitch provides a framework that brings contributors' applications to life and enables new users to have a variety of entry points for their individual adventures. In addition to exploring virtual reality with WebXR, creating inventive incremental games, and increasing the teaching of the next generation of digital natives in training programs and schools, Glitch is home to thousands of communities that are already growing (Glitch n.d.).

In order to use Glitch, developers can create an account using their GitHub, Google, or Facebook accounts. After that, a dashboard page is displayed for users to create a new project and store existing projects (Picture 19). Glitch offers some options for new projects, such as “glitch-hello-website” for a basic web page, “glitch-hello-node” for a simple Node application, “glitch-hello-react” for a new React project, Users also have the option to import an existing repository from GitHub (Picture 20).

The screenshot shows the Glitch dashboard with the following elements:

- Navigation:** Dashboard, Discover, Teams, Help Center, Blog, Upgrade.
- Header:** Glitch logo, search bar with 'New project' button.
- Main Content:**
 - Manage your projects:** Section with 'Project Hours' (1K / 12/1000 hours/month) and a 'Learn more' link.
 - Projects:** A table listing projects. The first project is 'supreme-spangle-savory', a simple Node app built with Fastify, instantly up and running. It has 13 project hours and was last edited on Apr 15, 2023.
- Starter apps:** A sidebar on the right with a 'Find More' link and a list of starter apps:
 - glitch-hello-website: Your very own basic web page, ready for you to customize.
 - glitch-hello-node: A simple Node app built with Fastify, instantly up and running.
 - glitch-hello-react: Get started with a new React project on Glitch!
 - glitch-hello-eleventy: Build a new Eleventy blog on Glitch!
 - glitch-hello-sqlite: Use a persistent SQLite database with your Node.js app.
- Footer:** Glitch logo, 'Build fast, full-stack web apps in your browser for free', 'Try something new' link, and various links for Company, Community, Create with..., and Social.

PICTURE 19. Glitch users' dashboard. Created projects are stored here and new projects can be created with the button on top right corner.

The screenshot shows the Glitch dashboard with a pop-up message overlaying the 'Manage your projects' section. The pop-up message is titled 'glitch.com says' and contains the text 'Paste the full URL of your repository' with a text input field containing 'https://github.com/orgname/reponame.git'. There are 'OK' and 'Cancel' buttons. The background dashboard content is partially visible, showing the 'Manage your projects' section and the 'Starter apps' sidebar.

PICTURE 20. Users can import projects from their GitHub. A pop-up message appears asking for the repository URL.

5 DISCUSSION

The goal of this project was to develop the internet platform known as 'Dream Blog', which would allow users to freely write and share their dream experiences as narratives. The project's objectives, which include the following, have been successfully met after lengthy development, modifications, and problem fixes:

- The project's user interfaces have been created to be straightforward and simple to use, ensuring that users enjoy a seamless writing and sharing experience.
- The application has been tuned to be usable on both PCs and mobile phones with the help of the Bootstrap framework.
- Users have the opportunity to write and share their stories anonymously, giving them the freedom and privacy to discuss their dreaming experiences without disclosing their identities.
- The front-end of the website has been successfully deployed on Netlify, making it reachable through <https://dream-blogs.netlify.app/>. Additionally, Glitch, which is embedded into the code of 'Dream Blog' for data storage and retrieval, has successfully been used to build and deploy the project's database, which can be viewed through this generated link: <https://supreme-spangle-savory.glitch.me/blogs>.

The application was developed utilizing the React development environment, which has enabled the project to carry out the necessary features and functions for an online writing and sharing stories website. In order to provide a better understanding, the usage of React is demonstrated in APPENDIX 1 for code explanation purposes. Overall, the project has succeeded in achieving its goals of developing a user-friendly and accessible platform for storytelling about dreams while also guaranteeing privacy and a successful deployment of the website and database components. Although 'Dream Blog' is not yet finished and might still have issues with its features and designs, further enhancements are planned and will be put into place soon in order to improve its overall performance and user experience.

REFERENCES

Beribey, Medium. 2020. Build and Deploy Website very very Simple with Glitch.com. Released on 09.11.2020. Read on 15.04.2023. <https://medium.com/coderes/build-and-deploy-website-very-very-simple-with-gitch-com-429a367d28e5>

Kendra Cherry, Verywellmind. 2022. Why Do We Dream? Updated on 14.10.2022. Read on 18.12.2022. <https://www.verywellmind.com/why-do-we-dream-top-dream-theories-2795931>

Anil Dash, Medium. 2018. What is Glitch? Released on 24.03.2018. Read on 30.03.2023. <https://medium.com/glitch/what-is-gitch-90cd75e40277>

Michael S. Franklin, Michael J. Zyphur. 2005. The Role of Dreams in the Evolution of the Human Mind. Journal of SAGE Publications Inc. Evolutionary Psychology 3(1). Released on 01.01.2005. Read on 14.04.2023. <https://doi.org/10.1177/147470490500300106>

Glitch. No date. Read on 15.04.2023. <https://glitch.com/about>

Martin Heller, InfoWorld. 2022. What is Visual Studio Code? Microsoft's extensible code editor. Released on 08.07.2022. Read on 21.03.2023. <https://www.infoworld.com/article/3666488/what-is-visual-studio-code-microsofts-extensible-code-editor.html>

Jordana A., Hostinger Tutorials. 2022. Why Should You Use Bootstrap. Released on 07.12.2022. Read on 14.04.2023. <https://www.hostinger.com/tutorials/what-is-bootstrap/>

David Herbert, Hubspot. 2022. What Is React.js? (Uses, Examples, & More). Released on 27.06.2022. Read on 21.03.2023. <https://blog.hubspot.com/website/react-js>

Javatpoint. No date. JSON Server. Read on 20.03.2023.
<https://www.javatpoint.com/json-server>

Björn Rasch & Jan Born. 2013. About Sleep's Role in Memory. Journal of the American Physiological Society. Psychological Reviews 93(2), 681-766. Released on 01.04.2013. Read on 14.04.2023. <https://doi.org/10.1152/physrev.00032.2012>

James Roland, Healthline. 2023. Why Do We Dream? Released on 13.04.2023. Read on 14.04.2023. <https://www.healthline.com/health/why-do-we-dream>

Catherine A. Sanderson, WondriumDaily. 2022. Why Do We Dream? Released on 05.06.2022. Read on 14.04.2023. <https://www.wondriumdaily.com/why-do-we-dream-2/>

Joel Varty, Agility. 2020. What is Netlify and What are its benefits? Release on 04.08.2020. Read on 30.03.2023. <https://agilitycms.com/resources/posts/what-is-netlify-and-why-should-you-care-as-an-editor>

Andrew Zola, TechTarget. 2022. What Is a Bootstrap and How Does It Work? Read on 20.03.2023. <https://www.techtarget.com/whatis/definition/bootstrap>

APPENDIX

Appendix 1. The use of React in coding and developing.

This appendix is written in order to explain the usage of React.js in the development of the 'Dream Blog' project. Some functions will be explained for more understanding regarding the code.

```
JS useFetch.js X
src > components > JS useFetch.js > [ⓧ] default
1  import { useState, useEffect } from 'react';
2
3  const useFetch = (url) => {
4    const [data, setData] = useState(null);
5    const [isPending, setIsPending] = useState(true);
6    const [error, setError] = useState(null);
7
8    useEffect(() => {
9      const abortFetch = new AbortController()
10
11      setTimeout(() => {
12        fetch(url, { signal: abortFetch.signal })
13          .then(res => {
14            console.log(res)
15            if (!res.ok) {
16              throw Error('Unable to fetch data from server')
17            }
18            return res.json()
19          })
20          .then(data => {
21            setData(data)
22            setIsPending(false)
23            setError(null)
24          })
25          .catch(err => {
26            if (err.name === 'AbortError') {
27              console.log('Fetch Aborted')
28            }
29            else {
30              setIsPending(false)
31              setError(err.message)
32            }
33          })
34        }, 500)
35      return () => abortFetch.abort()
36    }, [url])
37
38    return { data, isPending, error }
39  }
40
41  export default useFetch;
```

PICTURE 21. Custom React Hook 'useFetch' usage in useFetch.js, which will be used for other files.

A unique React Hook named 'useFetch' is in charge of using the 'fetch()' function to retrieve data from a specified URL (Picture 21). The URL of the data that needs to be fetched is the only argument that the useFetch method accepts. An object with three values is returned:

- 'data': data fetched from the given URL.
- 'isPending': a boolean value indicating whether or not the fetch process is active.
- 'error': displays anything that went wrong with the fetch process.

The real fetch request is sent using the 'useEffect' hook. It accepts a callback code as an argument, and this function is called whenever the value of the 'url' changes. To enable canceling the fetch process, a new 'AbortController' is created inside of this callback function.

The 'fetch()' function is executed with the supplied URL and an 'AbortSignal' object from the 'AbortController' following a 500 millisecond timeout. An error is thrown if the answer is not satisfactory. The 'setData', 'setIsPending', and 'setError' functions are then executed with the retrieved data, false, and null, respectively, unless the data is parsed as JSON. The error is caught and either logs a "Fetch Aborted" message (if it is an 'AbortError') or changes the 'isPending' state to false and the 'error' state to the error message.

Finally, if the component is unmounted or the 'url' value changes before the fetch operation is finished, the 'useEffect' hook returns a cleanup function that uses 'abort()' on the 'AbortController' to cancel the fetch operation.

Overall, using the 'AbortController' to stop unused requests and improve performance makes this custom Hook an effective method to handle data fetching in React components.

```

src > components > JS BlogDetail.js > BlogDetail
1  import React from 'react'
2  import { useParams, useNavigate } from 'react-router-dom'
3  import useFetch from './useFetch'
4
5  const BlogDetail = () => {
6    const { id } = useParams()
7    const { data: blog, error, isPending } = useFetch('https://supreme-spangle-savory.glitch.me/blogs/' + id)
8    const navigate = useNavigate();
9
10   const handleClick = () => {
11     fetch('https://supreme-spangle-savory.glitch.me/blogs/' + blog.id, {
12       method: 'DELETE'
13     }).then(() => {
14       navigate('/blogs')
15     })
16   }
17
18   return (
19     <div className="container-fluid">
20       <main className="wrapper" >
21         {isPending && <div style={{textAlign: 'center', padding: 400, fontSize: 32}}>Loading...</div>}
22         {error && <div>{error}</div>}
23         {blog && (
24           <article style={{padding: 100, display: 'flex', flexDirection: 'column'}}>
25             <h2 style={{textAlign: 'center', color: '#D4A017'}}>{blog.title}</h2>
26             <p style={{textAlign: 'center'}}> Written by {blog.author}</p>
27             <div className="blog-content">{blog.body}</div>
28             <button onClick = {handleClick} style={{textAlign: 'center', justifySelf: 'center'}}>Remove</button>
29           </article>
30         )}
31       </main>
32     </div>
33   )
34 }
35
36
37 export default BlogDetail

```

PICTURE 22. Stories detail is created via importing 'useParams', 'useNavigate', and React Hook 'useFetch' from react-router-dom with 'useFetch.js' module.

In Picture 22, the dynamic 'id' parameter, which will be used to fetch the particular blog post's details, is extracted from the URL (<https://supreme-spangle-savory.glitch.me> from Glitch) using 'useParams'. Using the 'id' parameter, the 'useFetch' hook is used to retrieve story data from a distant API address. When data is being fetched, the component displays a loading message ('isPending'), an error message ('error'), or the blog post information once the data has been fetched ('blog').

When a user hits the "Remove" button, a callback function called 'handleClick' is triggered. With the 'id' of the blog post to be deleted included, it makes a DELETE request to the same API endpoint. The 'useNavigate' hook directs the user back to the '/blogs' tab after a successful deletion.

```

src > components > JS AddNew.js > [⌘] AddNew > [⌘] handleSubmit
 1  import React, { useState } from 'react'
 2  import {useNavigate} from 'react-router-dom'
 3
 4  const AddNew = () => {
 5
 6      const [title, setTitle]= useState('')
 7      const [body, setBody] = useState('')
 8      const [author, setAuthor] = useState('Anonymous')
 9      const [isPending, setIsPending] = useState(false)
10      const navigate = useNavigate()
11
12      const handleSubmit = (e) =>{
13          e.preventDefault()
14          const blog={title, body, author}
15
16          setIsPending(true)
17
18          fetch('https://supreme-spangle-savory.glitch.me/blogs', {
19              method: 'POST',
20              headers: {"Content-Type": "application/json"},
21              body:JSON.stringify(blog)
22          }).then(()=>{
23              console.log('Added');
24              setIsPending(false)
25              navigate('/blogs')
26          })
27      }
  
```

PICTURE 23. “AddNew.js” - a React component allows users to create new stories.

By completing a form, users can make a new blog post using this “AddNew.js” React component (Picture 23). The form input values are tracked using the ‘useState’ hook, and page movement is handled using the ‘useNavigate’ hook.

Using the ‘useState’ function, the component first defines several state variables. These variables include ‘isPending’, a boolean that records whether the form is currently being submitted, ‘title’, ‘body’, and ‘author’, which store the values of the form input fields (Picture 24). The ‘useNavigate’ hook, which enables programmatic navigation to various application pages, is used to generate the ‘navigate’ variable.

When a user enters the given form, the 'handleSubmit' function is invoked. The default form submission behavior is prevented, a blog object is created with the input values, the 'isPending' state is set to true, and a POST request is made to the server to publish a new blog entry. The user is directed back to the "Stories Zone" page and the 'isPending' state is reset to false if the request is successful.

```

return [
  <div className="container-fluid">
    <main className="wrapper create">
      <h1>Create New Stories</h1>
      <form className="add-form" onSubmit={handleSubmit}>
        <label>Blog Title</label>
        <input type="text" required value={title} placeholder="Your Story Title" onChange={(e) => setTitle(e.target.value)}></input>

        <label>Blog Body</label>
        <textarea required value={body} placeholder="What is it that you are dreaming?" onChange={(e) => setBody(e.target.value)}></textarea>

        <label>Author</label>
        <span>You can use a nickname or keep it anonymous!</span>
        <input type="text" required value={author} placeholder="Anonymous" onChange={(e) => setAuthor(e.target.value)}></input>
        {!isPending && <button>Add Story</button>}
        {isPending && <button disabled>Loading...</button>}
      </form>
    </main>
  </div>
]
}
export default AddNew

```

PICTURE 24. Input form is generated in order to let users compose the story title, content and author of choice.

The return statement then displays the form's user interface. The original values of the input fields are determined by the 'title', 'body', and 'author' state variables, and are updated as the user types by the 'onChange' event handlers. The submit button is disabled and a "Loading" message is shown while the form is being completed using the 'isPending' variable.