BugSearch

Debug Your Bugs Instantly !

TEAM NAME: D-BUGGERS

TEAM MEMBERS:

RAJA KUMAR (2021CS10915)

AASTHA RAJANI (2021CS10093)

DISHA SHIRASKAR (2021CS10578)

ESHA PATEL (2021CS10566)

Overview

What is BugSearch ?

In this project, we aim to design a community-driven website that provides a platform for developers, programmers, IT professionals, and students to ask and answer technical questions related to software development and programming.

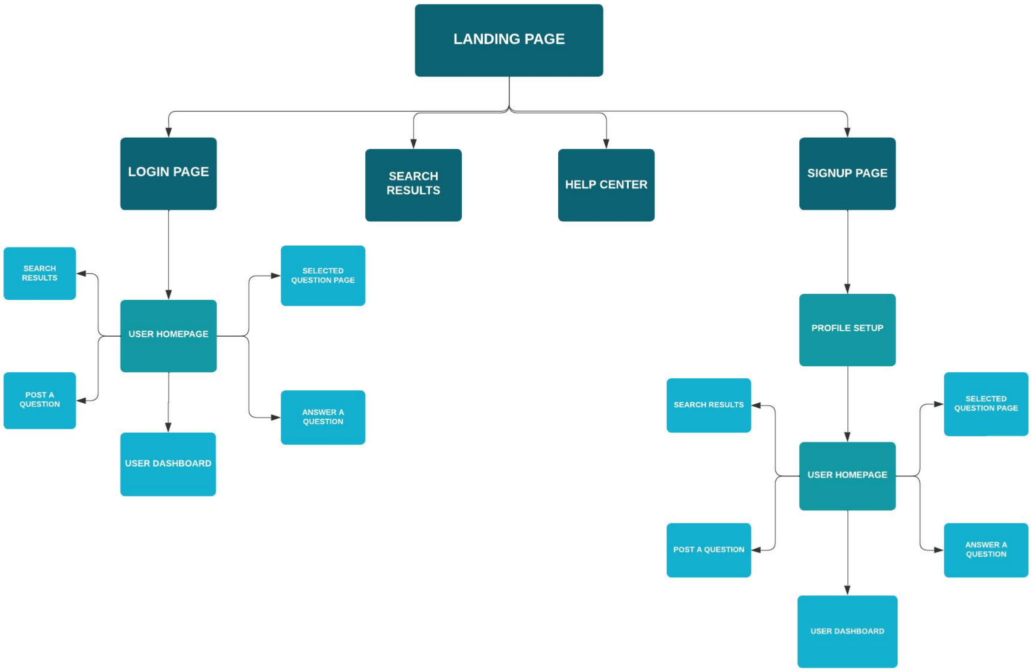
The name of our website is BugSearch. The website features a user-friendly interface where users can ask questions, post answers, and interact with other users. The site has a vast library of questions and answers that cover a wide range of programming topics, including programming languages, software frameworks, development tools, and many more.

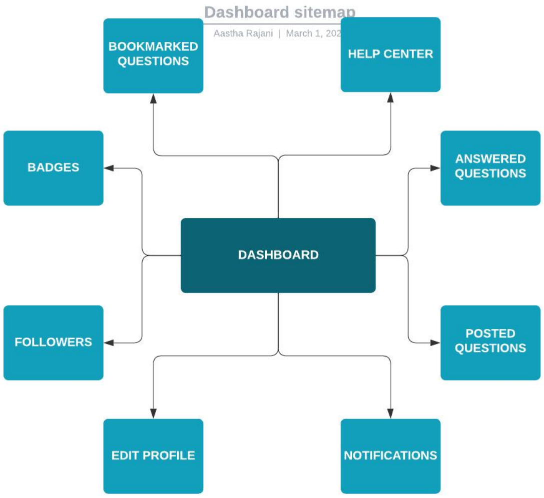
Overall, BugSearch is a great place for developers to find answers to technical questions, share knowledge, and build their online reputation in the programming community. Its user-friendly interface, tagging system, and interactive platform make it an essential tool for developers at all levels of expertise.

Why BugSearch ?

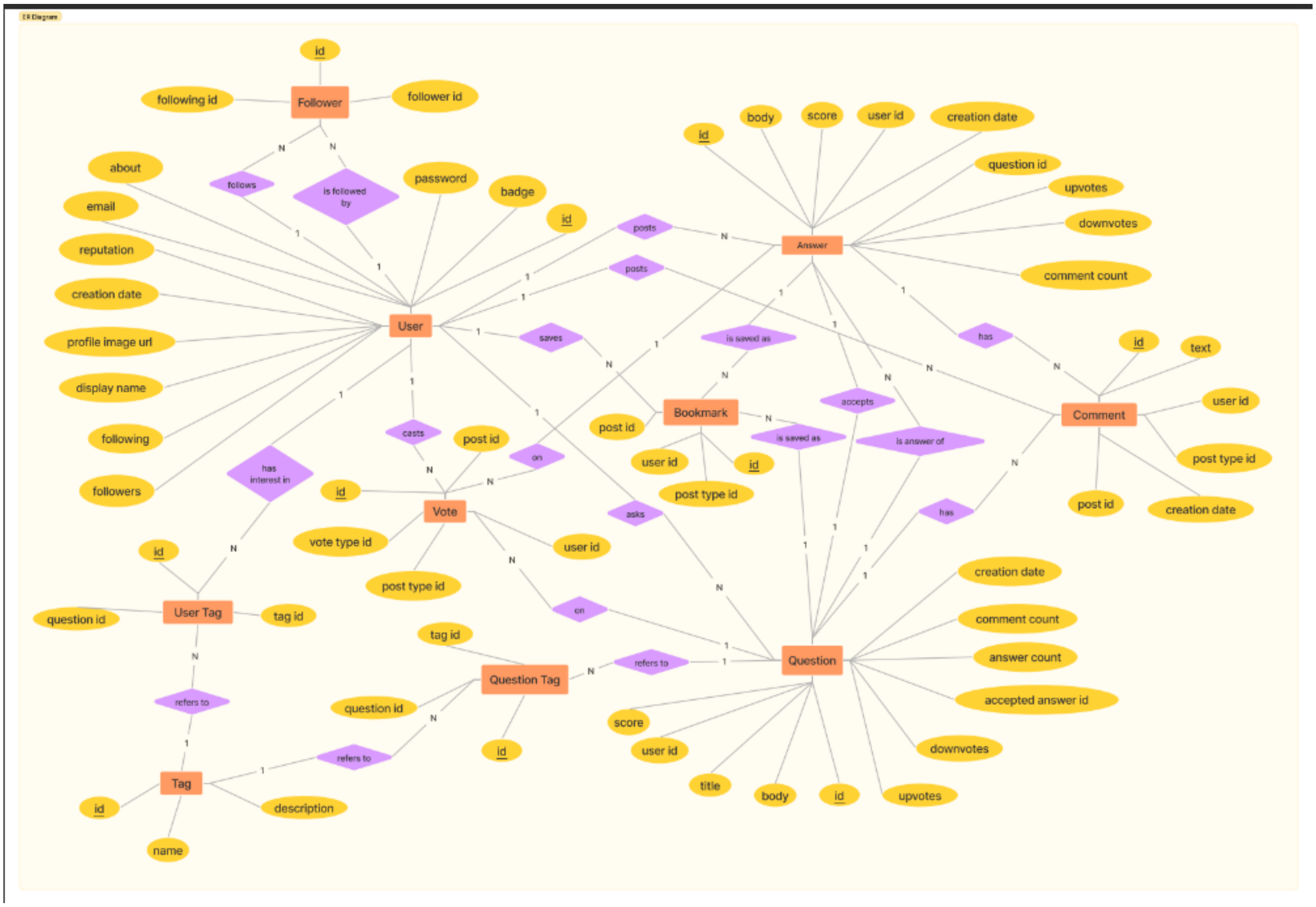
- **Specialisation**: While StackOverflow covers a wide range of topics, BugSearch focuses solely on coding-related questions. This could make it easier to find relevant information and receive specific help for coding issues.
- **Community**:While StackOverflow allows users to ask and answer questions, it does not have a built-in feature for users to follow each other or build a network of connections within the platform. By adding features like followers and following, developers might be able to build stronger connections and expand their networks within the programming community.
- **User Experience**: BugSearch may offer a more streamlined and user-friendly experience, with a more appealing user interface compared to StackOverflow, with features and interfaces designed specifically for coding-related questions.

Sitemap


LANDING PAGE
LOGIN PAGE
SEARCH RESULTS
HELP CENTER
SIGNUP PAGE
SEARCH RESULTS
USER HOMEPAGE
SELECTED QUESTION PAGE
POST A QUESTION
ANSWER A QUESTION
USER DASHBOARD
PROFILE SETUP
SEARCH RESULTS
SELECTED QUESTION PAGE
USER HOMEPAGE
POST A QUESTION
ANSWER A QUESTION
USER DASHBOARD


Dashboard sitemap
Aastha Rajani | March 1, 202
BOOKMARKED QUESTIONS
HELP CENTER
BADGES
ANSWERED QUESTIONS
DASHBOARD
FOLLOWERS
POSTED QUESTIONS
EDIT PROFILE
NOTIFICATIONS

ER Diagram


ER Diagram
id
following id
Follower
follower id
N
N
follows
is followed by
1
1
about
email
password
badge
id
reputation
creation date
profile image url
display name
following
followers
User
1
saves
N
posts
posts
1
1
N
id
body
score
user id
creation date
question id
upvotes
downvotes
comment count
Answer
1
1
N
has
N
is saved as
N
Bookmark
post id
user id
id
post type id
N
is saved as
accepts
N
is answer of
1
1
1
1
id
text
user id
Comment
post type id
post id
creation date
N
has
1
casts
N
post id
on
N
id
Vote
vote type id
user id
post type id
N
on
has interest in
N
1
id
User Tag
question id
tag id
N
refers to
1
Tag
id
name
description
1
refers to
N
tag id
Question Tag
question id
id
N
refers to
1
1
asks
N
Question
creation date
comment count
answer count
accepted answer id
score
user id
title
body
id
upvotes
downvotes

Links

Design Documentation :

https://docs.google.com/document/d/1BihoVlwISRzRCDUZkEgXo_icE6iT6nKWu5YLr9IiTmk/edit?usp=sharing_eil_se_dm&ts=63fe2f58

Yaml file for api design :

https://drive.google.com/file/d/1KKTtNab8ugY9vRnfQ-WqFnfQDKRbNgTX/view?usp=share_link

Figma Project Link :

https://www.figma.com/file/7KFxGA0vLIeYhf85DahLUR/E-Learning-(Community)?node-id=109%3A414&t=uU0vzFb7fXj2souA-1

Website Link :

http://10.17.51.150:5000

Test Coverage :

https://drive.google.com/drive/folders/1R_z7w3noXF0jrpF1zJvAnhvuVVnHQ6Dn?usp=share_link

ML-API Video:

https://drive.google.com/file/d/15T2mprrJlAZprThaROSCPPcaRiEXG8BC/view?usp=sharing

Github Link:

https://github.com/Eshapatel701/bug_search

Design Choices

Frontend Design Choices :

Tools used for designing Frontend : Figma, HTML, CSS, Javascript, JQuery, Bootstrap

- **User-Centred Design:** A user-centred design approach is critical for creating a website that meets the needs and expectations of its users. This involves understanding the target audience, their goals, and their pain points, and designing the website in a way that is intuitive and easy to use.
- **Responsive Design:** With the increasing use of mobile devices, it is important to ensure that the website is responsive, which means that it is able to adapt to different screen sizes and resolutions. This is important for providing a seamless user experience across different devices. We have tried to make some pages of our website responsive to meet such needs. Our website opens correctly even on mobiles to some extent.
- **Navigation and Information Architecture:** The website is well-organised and easy to navigate, with clear and concise labels for menu items and links. The information architecture is intuitive, with a clear hierarchy and structure that makes it easy for users to find what they are looking for. Sidebar and navbar have been used to make navigation through web pages easier.
- **Visual Design:** The visual design of the website is aesthetically pleasing, with a consistent and uniform colour scheme, typography, icons and layout. This can help to establish the brand identity and make the website more memorable.

Design Choices

Backend Design Choices :

Tools used for designing Backend : MySQL, MySQL_connector, Flask

1. **Database Design :** when designing the database, we had a few options for representing the data. One option was to use a single table to store all the information,while another option was to use multiple tables and establish relationship between them using foreign keys. We ultimately chose the latter option because it allowed us to organize the data more efficiently and reduce redundancies. For example, we created separate table for users, questions, answers, qustion_comments,votes,etc and established relationships between them using foreign keys.
2. **User Interface Design:** For the user interface, we had to decide on the layout and design the website. We had few options for the colour scheme , typography, and placement of elements. We ultimately chose a clean, minimalist design with a simple colour scheme and easy to read fonts . We also made sure to place important elements like the search bar and navigation menu in easy to find locations to improve the user experience.
3. **Voting System:** One important feature of Bugsearch is the voting system , which allows users to upvote or downvote questions and answers based on their usefulness. We had a few options for implementing this features, we have implemented a simple system where user can simply only upvote or downvote. It was easier to implement and provided a clear and straightforward way for users to express their opinions.
4. **Reputation System:** Another important features of BugSearch is the reputation system, which rewards users for contributing valuable content to the site. We had a few options for how to calculate reputation points for the user , we had use a simple system where users received the fixed number of points for upvote and downvote that will provide a sort of accurate way to look into user's activity level and

quality of their contributions, and allowed us to incentivize certain behaviours like providing helpful answers to difficult questions.

API Descriptions

Users API:

- '/users/dashboard' : Redirects the user to the profile page.
- '/users/complete_your_profile' : Fetches tags name and helps the user to update their profile.
- '/users/bookmarks' : Fetches bookmarked questions and answers of logged user.
- '/users/all_users' : Fetches all the users of the website sorted in alphabetical order of their name.
- "/users/followers" : Fetches all the followers of the user.
- "/users/following" : Fetches all the users whom the logged user following.
- '/users/help_with_login' : Redirects the user to help desk page .
- '/users/user_home' : Redirect to the homepage of user and fetches the details of the current logged user.
- "/users/posted_questions" : Fetches all the posted questions by the current user in sorted order of post time.
- '/users/trending' : Fetches the trending questions at current time.
- '/users/recommendations' : Fetches the recommended questions for the current user.
- ‘/users/search' : Fetches content based on the keywords in search.
- '/logout' : Helps logged user to logout .
- '/users/<string:username>' : Fetches the user based on their username.

API Descriptions

Posts APIs :

- '/users/questions' : Helps the logged user to post question.

- '/users/questions/<int:question_id>' : Fetches the question of that question_id and related answers, tags to this question and also upvote,downvote,bookmark status of question and its answers posted to this question

- '/users/questions/<int:question_id>/answers ': Helps user to post answer to the question with question_id as given in the url.

- '/users/questions/<int:question_id>/answers/<int:answer_id>/comments' : Helps user to post comment to the answers.

- '/users/questions/<int:question_id>/comments' : Helps user to post comment to question with question_id as given in the url.

Other APIs :

- '/updatefollow' : Helps the logged user to follow and unfollow any user and updating without reloading page
- '/getcomments' : Fetches the comments of post
- '/updatebookmark' : Helps the user to update bookmark status of the post
- '/updatevote' : Helps user to update the vote status of the post
- "/reset_password/<string:username>/<string:token>": Helps user to reset password through mail
- "/forgot_password" : Help user to redirect to forgot page
- '/help' : help page without user login
- "/login": Help user to login
- '/signup : Help user to signup in BugSearch

ML-API

We have used Eden-AI’s text moderation api to classify new comments as either acceptable or inappropriate.

Here's how it works:

Posting comments: The ML-API will analyse the text of the comment and determine whether it is acceptable or inappropriate.

Filtering out inappropriate comments: If the ML-API determines that a comment is inappropriate, a flash message will be displayed and the comment won’t be posted on the website. This helps to ensure that the website content is not offensive to any user.

Unique Features

- **Customizable user dashboard:** Users can customise their dashboard and edit their profile to display the types of questions or tags they are interested in, making it easier to find relevant content.
- **Advanced search functionality:** Advanced search functionality can help users find the most relevant answers to their questions by searching through various questions and topics of interest.
- **Personalised recommendations:** Usertags are used to provide personalised recommendations for questions, answers.
- **Reputation Points / Badges:** Providing an analytics dashboard for users can help them track their progress, see how their contributions are impacting the community, and identify areas where they can improve.
- **Question, Answer, Comment Posting :** Users are allowed to post comments , questions and answers on various topics.
- **Followers / Following :** Users are allowed to follow other users and may also be followed by other users.
- **Upvote / Downvote / Bookmark :** Users may upvote , downvote or bookmark a question to indicate their usefulness and relevance.

Other Features :

- **Login/SignUp:**
- **Password Reset: We have used python's flask-mail library to allow the user to reset the password. Once the user clicks on forgot password option, she/he will be redirected to a page where she/he will be required to provide his/her credentials based on which a jwt token will be generated and will be sent on the email_id provided. On clicking the reset link in the mail the user will be redirected to a webpage to reset password. This is completely secure as the token generated is unique and the reset password page can't be accessed without token.**

Challenges

1. **User Engagement :** Keeping users engaged and active on the platform can be challenging. So, we have addressed this issue by gamifying the platform

with badges, and reputation points, which incentivize users to participate and contribute.

2. **User-generated content :** Bugsearch relies on user-generated content, which can be of varying quality and accuracy. To ensure the accuracy and reliability of the content , Bugsearch allows multiple comment post and answer post by a user to a specific post.
3. **Quality control:** with a large number of user-generated content, maintaining quality control can be a challenging. Bugsearch uses a ml api which don't allow user to post abusive comments.
4. **Technical Challenges:** One of the main challenges in building a website is dealing with technical issues. This can include compatibility issues, server crashes, bugs, and other similar issues like history not being updated after reload, questions not getting bookmarked, software installation issues etc.Overcoming technical challenges requires a deep understanding of coding, debugging, and troubleshooting.

Limitations

- **Lack of code playground:** You can integrate a code editor or a code playground into your website using third-party tools such as CodeMirror or

Ace editor. This will allow users to write and test their code directly on the website without needing to switch to a different tool or environment.

- **Lack of support for multiple languages**: You can add support for multiple languages by integrating language-specific syntax highlighting and highlighting the keywords of different programming languages. Additionally, you can provide tags or categories for different languages to make it easier for users to find relevant content.
- **No option for uploading relevant images:** You can add the ability for users to upload images along with their questions or answers. This can be done by integrating a third-party file upload tool or using a cloud-based storage service such as Amazon S3 or Google Cloud Storage.
- **Less features for user without account :** Bugsearch doesn't offer many features to a user without an account. This could be done by adding additional features that make some basic features accessible to such users too.

As future work, we would try to work upon these points and many others in order to improve the user experience of our website.

	Tokens	Work done
AASTHA RAJANI (2021CS10093)	10	Frontend Design, Figma Prototype, Testing,

		Documentation
DISHA SHIRASKAR (2021CS10578)	10	Frontend Design, Figma Prototype, Testing, Documentation
RAJA KUMAR (2021CS10915)	10	Login/ Logout/Signup api, Database design, api for voting system, api for follow feature, api for bookmarks, api for reputation system and other api design, integrating backend to frontend, object class implementation, Swagger API design
ESHA PATEL (2021CS10566)	10	ER Diagram, api for search tool, api for password recovery, api for image uploading, database design, ml-api, uploading on baadalvm and setting up database