Feature/Icon request interface for EOS-Icons [Front-end and UX].

Org - Python Software Foundation
Sub org - EOS Design System

---

Name: Devesh Vijaywargiya
Mail: <dvijaywargiya@gmail.com>
Gitlab: <https://gitlab.com/dvijaywargiya>
Linkedin: <https://www.linkedin.com/in/devesh-vijaywargiya>
Timezone: <UTC +5:30>
Motivation

Now that I've decided to pursue my career in the Software domain, I want to learn how a Software is actually built; how so many people are able to work together and collaborate, and how to quickly and efficiently write production level code.

As of now, I've worked only on small projects, with small teams focussing on one technology at a time, so this will be a completely new experience for me and I hope to learn a lot of new things.

Mentors: Cynthia Sanchez, Sorin Curescu, Kartikay Bhutani, Abhinandan Sharma, Jesus Herman

Synopsis

I am going to create a web interface for the EOS community where users will be able to request for new icons (and possibly features) in an elegant and interactive manner. I think the EOS team will like this project because the existing way to do this job is not very user friendly, because of which they are possibly missing out on good ideas from people around the globe. This web interface can potentially help them reach a wider audience. I have worked on UI projects in the past, so I know how time taking and repetitive it can become. I've spent several hours on a single component to make sure it looks perfect. I'll work with the same determination on this project as well.

The fact is, I like web development and want to pursue a career in the same field, so I have a constant motivation to learn new things in this direction. Also, I'm one of those people who likes getting things done on time. I hate delaying work. You can be rest assured, I'll try my best to never miss a deadline and complete this project on time with all the requirements.

EOS - Design System

EOS is a community based open source project for building a customizable Design System that can help open source organizations, SMEs and others deliver outstanding user interfaces and consistent user experience across-products.
**Project Background**

The EOS team is looking for a way to make it easier for users to be able to request new ICONS into EOS. Currently, users have to go and create an issue in their Gitlab repository, which is not a very elegant and user friendly way.

**Proposed Idea and Project Goal**

- The Idea is to create a web interface and incorporate it into the existing EOS website where a user will have the ability to request for new icons (or possibly features) after signing up.
- To ensure that the community has a way to contact each user, every user will have to provide a valid email-id while signing up. Other than email-id, users can provide their Gitlab and/or Github handle, Phone number, Linkedin profile, etc. Will confirm which ones to use with the mentor.
- Users will also have the option to look at requests from other users of the community as a part of their feed. They will be able to vote for a request and/or have a discussion on it.
- Discussions will be comments based, where a user can reply to other users and form a thread.
- Non-authenticated users will be allowed to see the feed but they will not be able to interact with it (vote/comment).
- There will also be a few designated users with admin access. They will be able to do everything that a normal user does plus have additional rights.
- Additional admin rights can possibly include:
  - Ability to view the submitted requests, review them, have a discussion on them with other administrators and choose whether a particular request will be published for the community or not.
  - Change status of a request. Here, status can follow below sequence:

    \[Submitted \rightarrow Under\ Review \rightarrow Published \rightarrow Under\ Community\ Review \rightarrow Rejected/Accepted\]

    - A `state` will determine whether a user is authenticated or not. It will also tell us whether the given user is an administrator or a normal user.

**Proposed Technology Stack**

- Since this Project mainly involves building of the Frontend / UI-UX for the said web interface, I plan to use React JS as the tool.
The reason I've chosen React JS is because I'm familiar with function components in React and also I'm well versed with Context and React Hooks to achieve a seamless functionality in a web application.

For styling I plan to use SCSS and write it myself for all individual components. I'll use SCSS over conventional CSS because it allows features like variables, nestings, inheritance etc, which helps in writing concise and distributed code.

Sample Project Pipeline

- Change status of a Request
- View all submitted requests and have a discussion
- Admin Home Page
- Complete Access
- Admin Authentication
- Log Out
- Landing Page
- View Access Only
- User Authentication
- User Home Page
- Make a new Request
- Requests Feed

Only authenticated users (both Normal and Admin) can vote and discuss
Workdone and Research <Link: Frontend, Backend, Demo>

- As discussed with the mentors, one possibility for the backend is to use Strapi CMS, so I tried working with it and have successfully created a small Web Application (Frontend, Backend) with a few functionalities.
- Prototype Login/Signup Page

![Login/Signup Page](image1)

- Prototype User Homepage

![User Homepage](image2)

- Strapi requires an instance of MongoDB for running. Here I am running a locally hosted version for the backend. For the project, the backend will be hosted online. One possibility for hosting is to use Heroku. It also provides free addons that can be used for the MongoDB requirement.
- **Authentication State Management:**
  - I’m maintaining the state of the application (authenticated/not authenticated) by using a parent context which can be accessed from all the subsequent pages. I’ve used `useReducer` to update this state.
  - Using `useReducer` over `useState` because state change is needed to be performed non-locally. `useState` is better suited when states are changed locally to a component.
  - Our state will also tell us whether an authenticated user is a normal user or an admin.

- **API Calls and Backend:**
  - For the purpose of Login I used a Strapi API to send a POST request consisting of an identifier and a password for authentication. On successful authentication, a JWT token is returned.
  - For the purpose of Signing Up I used a Strapi API to send a POST request consisting of Email, Username and Password (for now, can be extended). On successful authentication, a JWT token is returned.
  - As discussed, enabling CORS in the Strapi backend will help us achieve the required security for the application.
  - I’ll be using Axios for communicating with the server. It is a client HTTP API based on the XMLHttpRequest interface provided by browsers. Advantages of using Axios over Fetch:
    - Responses and requests can be intercepted.
    - It has a streamline error handling mechanism.
    - It has the ability to cancel requests.
    - Support for upload progress.
  - APIs for different request related to authentication:
    - Registration
      ```javascript
      axios
      .post('http://localhost:1337/auth/local/register', {
        username: 'Strapi user',
        email: 'user@strapi.io',
        password: 'strapiPassword',
      })
      ```
    - Login
      ```javascript
      axios
      .post('http://localhost:1337/auth/local', {
        identifier: 'user@strapi.io',
        password: 'strapiPassword',
      })
      ```
- Forgot Password

```javascript
axios
  .post('http://localhost:1337/auth/forgot-password', {
    email: 'user@strapi.io',
  })
```

- Email Validation

```javascript
axios
  .post('http://localhost:1337/auth/send-email-confirmation', {
    email: 'user@strapi.io',
  })
```

- Password Reset

```javascript
axios
  .post('http://localhost:1337/auth/reset-password', {
    code: 'privateCode',
    password: 'myNewPassword',
    passwordConfirmation: 'myNewPassword',
  })
```

- [http://localhost:1337](http://localhost:1337) will be replaced by the server address.*

- Comments and Discussions:
  - I’ll be implementing “mentions” enabled discussion forum. For “mentions”, I looked up [rc-mentions](https://github.com/raulcnrs/rc-mentions). I was able to incorporate such a feature in my prototype.
  - I’ve added a minimal textArea in the prototype. As of now, you can mention any of the registered users (using “@” as a trigger).
  - I am working on what is the best method to store these comments in the database. Will talk with the backend intern about the same if I get selected.

- User Interface:
  - I went through other platforms that offer a similar interface and have come up with a sample version of my own. I have attached a few screenshots under the “Proposed UI” heading.
  - I’m keeping simplicity and elegance as the primary goal for this web interface.
- Prototype - All Requests Page

- Prototype - New Request Page
Prototype - Page for every request

**EOS Icon Request**

**Title1 LoremIpsum**

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum. Why do we use it? It is a long established fact that a reader will be distracted by the readable content of a page when looking at its layout. The point of using Lorem Ipsum is that it has a more-or-less normal distribution of letters, as opposed to using ‘Content here, content here’, making it look like readable English. Many desktop publishing packages and web page editors now use Lorem Ipsum as their default model text, and a search for ‘Lorem Ipsum’ will uncover many web sites still in their infancy. Various versions have evolved over the years, sometimes by accident, sometimes on purpose (e.g. a fly introduced humour and the like).

Votes: 20 - 5

Start mentioning with @

**Proposed UI**

[Link: Sample UI]

**Current Interface**

On click of the “Request new icon” button, the user is redirected to the Gitlab repo and asked to create an issue for their request.
Sample User Homepage

This is a very simple example of what the homepage can look like. Non-authenticated users will be able to look at requests made by other users and view details about them.

Sample Admin Homepage
Sample Dropdown

Welcome to EOS Icons
Look at what's going on in our community

<table>
<thead>
<tr>
<th>Categories</th>
<th>Status</th>
<th>Votes</th>
<th>Threads</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Submitted</td>
<td>20</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>Under Review</td>
<td>23</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td>Published</td>
<td>33</td>
<td>27</td>
</tr>
<tr>
<td></td>
<td>Community Review</td>
<td>21</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>Rejected</td>
<td>6</td>
<td>22</td>
</tr>
<tr>
<td></td>
<td>Accepted</td>
<td>15</td>
<td>22</td>
</tr>
<tr>
<td></td>
<td>Accepted</td>
<td>15</td>
<td>18</td>
</tr>
</tbody>
</table>

Sample Authentication (Login)

Welcome to EOS Icons
Look at what's going on in our community

<table>
<thead>
<tr>
<th>Categories</th>
<th></th>
<th>Votes</th>
<th>Threads</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title 1</td>
<td></td>
<td>20</td>
<td>18</td>
</tr>
<tr>
<td>Title 2</td>
<td></td>
<td>23</td>
<td>25</td>
</tr>
<tr>
<td>Title 3</td>
<td></td>
<td>22</td>
<td>27</td>
</tr>
<tr>
<td>Title 4</td>
<td></td>
<td>21</td>
<td>18</td>
</tr>
<tr>
<td>Title 5</td>
<td></td>
<td>6</td>
<td>22</td>
</tr>
<tr>
<td>Title 6</td>
<td></td>
<td>15</td>
<td>22</td>
</tr>
<tr>
<td>Title 7</td>
<td></td>
<td>15</td>
<td>18</td>
</tr>
</tbody>
</table>
Sample User Homepage

The user can view his requests through the “My Icons” button and request a new icon using “Request a new Icon” button.

Sample New Request Page

Users can request for a new Icon using the following form.
**Sample Request Page**

How an individual request will look to a user.

- Will add/work on more sample pages based on mentor feedback.
Timeline

**BONDING AND DISCUSSIONS**
Solidify the requirements and improve upon this timeline.
Continue working with the EOS team and contribute to issues/features in React Migration/other ongoing projects.
Discuss and finalise the Tech Stack with the mentor.
Get comfortable with the primary mentor and mutually decide on a daily/weekly meeting schedule for the entire duration.
Discuss and improve on the sample UI based on discussion with the mentor and the backend intern which will be used as primary reference while writing code.

**PHASE 0: COMMUNITY BONDING**

**JUNE 1**

**WEEK 1: AUTHENTICATION**
Work on the front-end for authentication
Log In, Sign Up, Password Recovery (Styling included)

**WEEK 2: ROUTING**
Set up routes and make API requests.
Write logic for password recovery and test it.

**WEEK 3: TESTING**
Test the working of this entire sub unit of the application and give a demo.
Decide on the different frontend routes (Request New, Admin Panel etc.) and write their baseline code.

**WEEK 4: FRONTEND**
Discuss and solidify user feed features.
Work on the Frontend for the user feed and make API requests.

**EVALUATION 1**
WEEK 5: REQUEST NEW ICON
Incorporate the NEW ICON feature in the interface.
Add "My Requests" option for an authenticated user to view his/her requests and their current status.

WEEK 6: VOTING
Discuss about the best way to go about this feature with the Backend Intern and work on this.
Set up Frontend and API requests for the same.

WEEK 7: COMMENTS AND DISCUSSION
Implement the "mentions" enabled COMMENTS feature.
Work with the Backend intern to realise the best method to store the comments in the backend.

WEEK 8: ADMIN RIGHTS
Develop Admin Homepage and set up routes to different Admin rights.

WEEK 9 AND 10: SET UP API CALLS TO THE BACKEND
Start setting up the API calls as discussed with the Backend Intern.

WEEK 11 & 12: BUFFER AND ADDITIONAL FEATURES
Use this time as a buffer period.
Work on additional features as discussed with the mentor.
About me

I am a junior at International Institute of Information Technology, Hyderabad pursuing Bachelor’s degree focussed in Computer Science and Engineering. I’ll be completing this degree by April 2021.

I love outdoor sports and try my best to take part in all outdoor activities in my college. This also means that I get injured a lot, but it is all part of the fun. Particularly, I love long distance running and playing Football. I play as a Central Midfielder primarily. I am a Real Madrid and CR7 person.

I also indulge in a bit of Computer and Mobile Gaming in my free time. Counter Strike tops my list of competitive games. I want to travel around Europe before I turn 25, so if I get selected and successfully complete this program then I’ll save my earnings for such a trip.

Talking about PSF and EOS, I chose PSF because I thought it would be good for my profile as a web developer if I am associated with Python Software Foundation. I chose EOS specifically because I found the community quite approachable (thanks to Slack). Also, I liked the fact that I’ll be working with React as a part of this GSoC project. I want to get well versed with this framework and I think this project will help me do that.

Work Experience

- My Resume <Resume>
- Currently, I am working as a R&D Intern at Indian School of Business, Hyderabad. My work here involves web scraping, web development and data analysis. I like to keep myself busy, so I often take up projects like these. I work remotely, with flexible timings and a flexible contract. If I am selected for GSoC, I will withdraw myself from this project.
- Before this, I was working for Virtual Labs, India for about a year. I was initially an individual contributor there and then worked as a Product Development Intern with 15 junior interns reporting to me. My work there revolved around web development using Vanilla Javascript (and a few custom libraries for visualisations). There, I learned about effectively using version control and the difference between writing code for college assignments and writing production level code. I was also responsible for the work done by the 15 interns under me. I reviewed their codes and was responsible to make sure that they follow and complete their milestones on time.
Open Source Contributions

I developed a knack for web-development over last summer; so I tried learning new things and came to know about how open source coding works. A friend of mine introduced me to the world of open source. I initially made a few contributions in Gnome and then I started working in EOS-Design.

- EOS-Design System
  - All Merge Requests (Open/Closed/Merged)
  - All issues created
- Gnome
  - https://gitlab.gnome.org/GNOME/epiphany/-/merge_requests/540
  - https://gitlab.gnome.org/GNOME/gnome-maps/-/merge_requests/73
  - https://gitlab.gnome.org/GNOME/gnome-maps/-/merge_requests/68

As a part of my year long internship at Virtual Labs, India I worked on building an educational platform. The aim of Virtual Labs, India is to build a platform to be used across all colleges in India. Me and my partner developed two experiments in the Data Structures module from scratch. These will be made open source when the entire platform is built. If you want to take a look at my work, you can look in the Commit histories.

- Bubble Sort (Link to Demo)
- Selection Sort (Link to Demo)

Commitments during GSoC period

As already mentioned above, if I get selected then I’ll withdraw myself from my project at the Indian School of Business. I’ll be available full time and hence will be able to devote at least 30-35 hours a week.

As of now, I don’t have any other commitments during the GSoC period. If something does come up, I’ll discuss it with my mentor. In any case, I am used to working when I’m travelling so the number of workless-days will be absolutely minimal.

THANK YOU FOR READING