LABELLAB

Google Summer of Code 2019

Mentor: Oshan Mudannayake

Personal Information

Basic information

Name: Tushar Varshney
Gender: Male (he/him/his)
Nationality: India 🇮🇳 | IST (UTC + 5:30)

Contact information

Email address: tvarshney@me.iitr.ac.in
Phone number: +91 7302201561
Emergency phone: +91 9424062086

Educational information

Institute: IIT, Roorkee
Degree: B. Tech
Major: Production and Industrial
Graduation: 2021

Social media

GitHub: Tushar19varshney
LinkedIn: Tushar Varshney

About me

1. Are you a SCoRe contributor/ Have you contributed to SCoRe before?

Yes, I am SCoRe contributor, following are my pull requests and opened issue:

- **Pull Requests / Patches:**

  #13 (**Merged**) : Restructured and reformatted project with eslint and standard js

  #28 (**Merged**) : in db/dump/stories.json object's field featured_image_url changed to featured_img_url

  #88 (**Merged**) : Flask backend

  #115 (**Merged**) : Dockerfile added for flask and updated docker-compose.yml

  #135 (**Merged**) : Authentication tests and Fetch stories test

  #148 (**Merged**) : .travis.yml file added with python versions(3.5 and 3.6)

#151 (**Merged**) : assert checks updated in Flask API's tests

#168 (**Merged**) : CORS added

#175 (**Merged**) : Register API bug fixed and docstring added

#176 (**Merged**) : Test bugs fixed as API's got updated

#197 (**Merged**) : user_name added in jwt_token payload

#204 (**Merged**) : error handling of register and login forms, proxy port changed as flask default port

#230 (**Merged**) : structurize the __init__.py file

#234 (**Merged**) : fix: generalized the text on OAuth buttons

#248 (**Merged**) : Fixed the console warnings

#262 (**Merged**) : feat: add elasticsearch install and starting the server instructions

#264 (**Merged**) : fix: README.md updated

#266 (**Merged**) : docker-compose integrated in travis build

- **Issues:**

#12 (**Closed**) Restructure and reformat project

#27 (**Closed**) Unit tests for fact-bounty-server

#80 (**Closed**) 'feartued_image_url' field should be changed to 'featured_img_url' in 'db/dump/stories.json'

#113 (**Closed**) Allow CI/CD mirroring to handle dependency checks

#104 (**Closed**) Tests for flask API's

#145 (**Closed**) Docker file for fact-bounty-flask

#150 (**Closed**) Assert checks in unittest of Flask API's

#174 (**Closed**) Register API issue

#192 (**Closed**) Standard Project Proposal template for GSOC

#194 (**Closed**) dispatching auth action found undefined error and token variable error

#196 (**Closed**) token payload user data missing

#203 (**Closed**) error handling of Login and Register form , proxy port should be changed to default flask port

#233 (**Closed**) Text on OAUTH buttons is not correct

#247 (**Closed**) Console warnings

#261 (**Closed**) Instructions for Elasticsearch installation and starting the server

2. How can we reach you (eg: email) if we have questions about your application?

Email - tvarshney@me.iitr.ac.in

IRC(freenode) / Gitter - Tushar19varshney

3. What is your GitHub username(s):

Username - Tushar19varshney

Github - https://github.com/Tushar19varshney

Project Specific Questions

4. Which SCoRe GSoC project are you applying for :

I want to apply for **Labellab**: Machine learning supported web-based image labelling tool for the researcher.

5. What do you plan to accomplish over this summer for this project? (Please tell us) :

LabelLab is an image analyzing and classification platform. The web application should allow users to upload batches of images and classify them with labels. It will also have the features to run classifications against a trained model. LabelLab also has a user project management component as well as an image analyzing component. I have listed down the basic layout of Labellab that to be implemented during **GSoC** below:

Milestones:

- Basic authentication system for user consisting OAuth support of Google, Facebook and Github.
- **Label Hub:** Dashboard for maintaining the user activity, creating new projects, changing settings related to the user profile or labelling tool.
- User can upload a single as well as batches of the image after creating a new project.
- **Labeler:** A Labelling tool which can label objects in the image and all the data get stored in the database.
- ML model can be trained using these stored images and which can provide the necessary output for a new image.
- **Label Analytics:** Analytics and summaries regarding the uploaded images and batches should be available to the users in the form of line graphs, pie charts or histograms on the basis of certain image processing parameters. Analytics like average number of

objects in an image, objects that are marked more oftenly, classes of objects, objects positioning in an image and many more can be analyzed with graphs and charts.

- **Path Tracking feature:** It has a function to state the location of the images that were taken. It can be implemented by using Exiftool, a free and open-source software program for reading, writing, and manipulating the image, audio, video, and PDF metadata. When a single unique object is labelled over multiple images, the platform should be able to generate the path of the movement of the object.
- User guide for using the Labelling tool which will give relief to the researchers in using the tool.
- Wiki documentation for deploying the platform in user's own server space.
- Writing documentation on Sphinx, a tool that makes it easy to create intelligent and beautiful documentation.
- **Dockerizing** the Labellab for Reproducibility, Environment Management, Continuous Integration.
- Integrate the build and test cases to **Travis CI**.

If time permits I would like to implement some ideas listed below:

Extra Milestones:

- The user can prepare a custom report of the project according to the template provided in the app on the basis of analytics and results.
- The segment showing all the labelled objects in all the images uploaded by a particular user on the Dashboard.
- In path tracking, we can further add access to the webcam and allow videos upload for detecting the flow of object.
- Static Page of LabelLab which display all the related information of Labellab and a get started button which will direct user to signup and user can log in to Labellab.
- Implementing project handling by adding members to the project. The project can be executed within teams.
- Multi-language support for giving user a good experience.

6. If you have your own project to propose, please describe it here :

Till now, I don't have any project idea but I want to implement some extra features which I have already mentioned in the above section, along with the main project, within or after GSoC official timeline.

7. Projects related details. (Have you tried that project you selected from SCoRe project list? What problems, if any, were presented? What prevented you from getting the entire system up and running?) :

I have successfully completed the task for implementing the basic prototype of Labellab which consist of Basic authentication, Image Labelling tool, Continuous Integration, Dockerizing and also implemented testing. I have faced some feature implementation problems but overcame with the help of the mentor and community.

8. List down any plans you have during this summer (over the time period of GSoC, such as classes, job, vacation plans, thesis, etc.) :

My vacations start on 5th May 2019 and end on 11th July 2019. The official GSoC period is from 6th May 2019 to 26th August 2019. I can easily devote 50-55 hours per week until my college reopens and 45-50 hours per week after that. Other than this project, I have no other commitments during the summer vacations. I shall keep my status posted to all the community members on a weekly basis and maintain transparency in this project.

9. Tell one interesting fact about yourself :

I enjoy getting *400's* in my logs rather than *200's*.

10. Few sentences about the student and why he/she thinks as the best person to do this project :

I have a keen interest in software development since my freshman year. In my pursuit of exploring this field and enhancing my knowledge, I became an active contributor to open source and ScoreLab community. I have been contributing to ScoreLab since February and my journey so far with has been a great one. Every pull request I have made to ScoreLab has taught me something new and interesting. I have sound knowledge of Python, NodeJs, ReactJs, Redux, RESTful APIs and databases which is the main technological requirement of the LabelLab project.

Some General Questions

1. **Education:** 2nd year - 2nd-semester B.Tech, Production and Industrial Engineering (Mechanical Department), IIT Roorkee, India
2. **Programming Experience:**
    - Basic knowledge of C++
    - Sound knowledge of OOPS and a good understanding of MVP architecture.

- Server Side:
    - PHP
    - Python (worked with Django and Flask)
    - Node.js
    - Good understanding of REST API and network calls
- Database Systems known:
    - MySQL
    - SQLite
    - PostgreSQL
    - MongoDB (mongoose as ODM)
- Front End Development:
    - HTML5, CSS3, Bootstrap
    - Javascript (Vanilla, ReactJS)
    - Redux (State management tool)
    - jQuery
- Also known Chrome Extension and Slack Bot Development

3. **Development Environment**
    - Ubuntu 18.04.2 or Windows 10
    - Visual Studio Code as a text editor
    - Good familiarity with virtualenv.
4. **Version Control**
    - Top Preference: Git (very strong concepts)
5. **Projects**

   I am a programmer in the learning phase, attending hackathons and developer summits. Here, I have listed some interesting projects that I have done as class projects, hackathon projects and college projects:

    - **Thomso'18: Thomso** is the Annual Cultural Festival of IIT Roorkee. I have been part of the web development team of **Thomso'18** and successfully planned and implemented the stagger website. Gathered around 1M hits around the globe; Reached 25,000 users in around 2 years.

      Stack- Node.js , React.js, MongoDB

      GitHub- ThomsoIITR

      Visit- www.thomso.in

- **Exame:** Successfully developed examination portal with face login/register, live invigilating with the webcam, and admin interface for setting up the question paper.

  Stack- Node.js , React.js, MongoDB

  GitHub- **Tushar19varshney/Exame**

- **Entrepreneurship Cell:** Ecell is the entrepreneurship cell of IIT-Roorkee. I am Technical Operational Manager at Ecell, IIT Roorkee and developed the Ecell website from scratch.

  Stack- Django, PostgreSQL, React.js

  Visit- **www.ecelliitr.org**

- **Esummit:** E-Summit is the annual entrepreneurship summit of IIT Roorkee and successfully developed and implemented the E-Summit website.

  Stack- Django, React.js

  Visit- **www.esummit.in**

- **File Manager:** Developed front end of file manager web app of Channeli, the intranet website of IIT Roorkee students.

  Stack- React.js, Redux

There is a number of personal projects I have worked on and also contributed to the development endeavours of others. All of that can be found at my **GitHub profile**.

Project Introduction

LabelLab: The basic requirement for the LabelLab project originated with the need to label thousands of images of elephants. However, the project was generalized to label not just elephant images, but any animal in general.

Proposed workflow: To use LabelLab web application user can log in either by credentials or get authenticated using OAuth. The application will have Oauth provided by Google, Facebook and Github. If a user is an existing one he directly comes to the home page. After user get authenticated user will land on the dashboard.

The dashboard will consist mainly of two sections: start a new project and previous projects or recent activities. When a user starts a new project a labelling tool appears whose basic function is to mark objects in the images. On recent activities section, all the previous projects of the user will appear and on clicking the card user can view analytics like graphs, results, etc and can edit the project.

Labeler (Labelling Tool): It is a tool which can classify the objects by making a box around it using the mouse(see the gif). This tool will be implemented using HTML canvas and javascript mouse events. The user can upload single or multiple images and on submitting the labelled images all the data will be stored in the database.

Path Tracking feature will be implemented in the number of images using **ExifTool,** a tool which captures GPS coordinates and stored them as “metadata” which will be embedded in the photo.

Rest API’s will be implemented in Node.js for executing server requests and mongoose will be used as MongoDB ODM for querying to the database.

This stored data can be used to train the ML model. User can switch or choose from multiple ML models for testing their projects samples. As mentioned in the document, we don’t have to build ML models, we can use pre-built models of Tensorflow, Keras, Theano or Microsoft Cognitive kit for testing the samples. Once the ML model gets trained, its predictions can be utilized on the new submission of images. So next time when the user uploads the images, labels will be assigned to the images automatically.

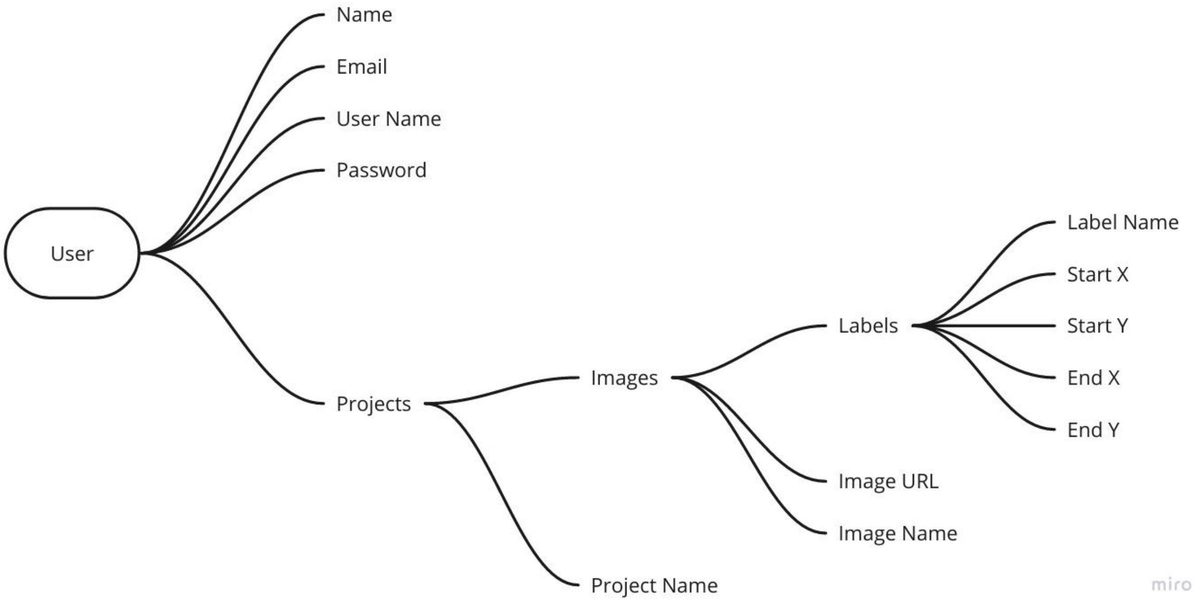
Project Goals

- User authentication for Labellab either by credentials or OAuth.
- REST API implementation in Node.js and Express.js and testing of API’s using Unit tests.
- Dashboard for maintaining user activity and changes.
- Image labelling tool in which one or many images can be uploaded and the user can mark objects in the image and all data stored in the database.
- User guide for using the Labelling tool.
- Analytics and summaries of the uploaded images of the user.
- Path Tracking feature: should have the function to state the location of the images that were taken.
- User can switch or choose from multiple ML models for testing their projects samples.
- Basic UI using **Semantic-UI** library for ReactJS
- Dockerization of both Labellab frontend and backend.
- CI/CD integration in **Travis-CI.**
- Wiki documentation to help the user in deploying the user’s own server space.
- Documentation of Labellab using Sphinx

Implementation

1. Multi-channel Authentication system

- Initialize the backend server
    - Build a **REST API** server on Node.js, with Express.js as the web framework, which shall handle the requests and query made by the users to the database.
- Integrate a NoSQL(non-relational database)
    - MongoDB provided the best non-relational database management system for a server set up on Node.js. Its ODM, Mongoose, is a layer on top of MongoDB which ease the process of monitoring, managing and process huge volumes of data in a systematic way.



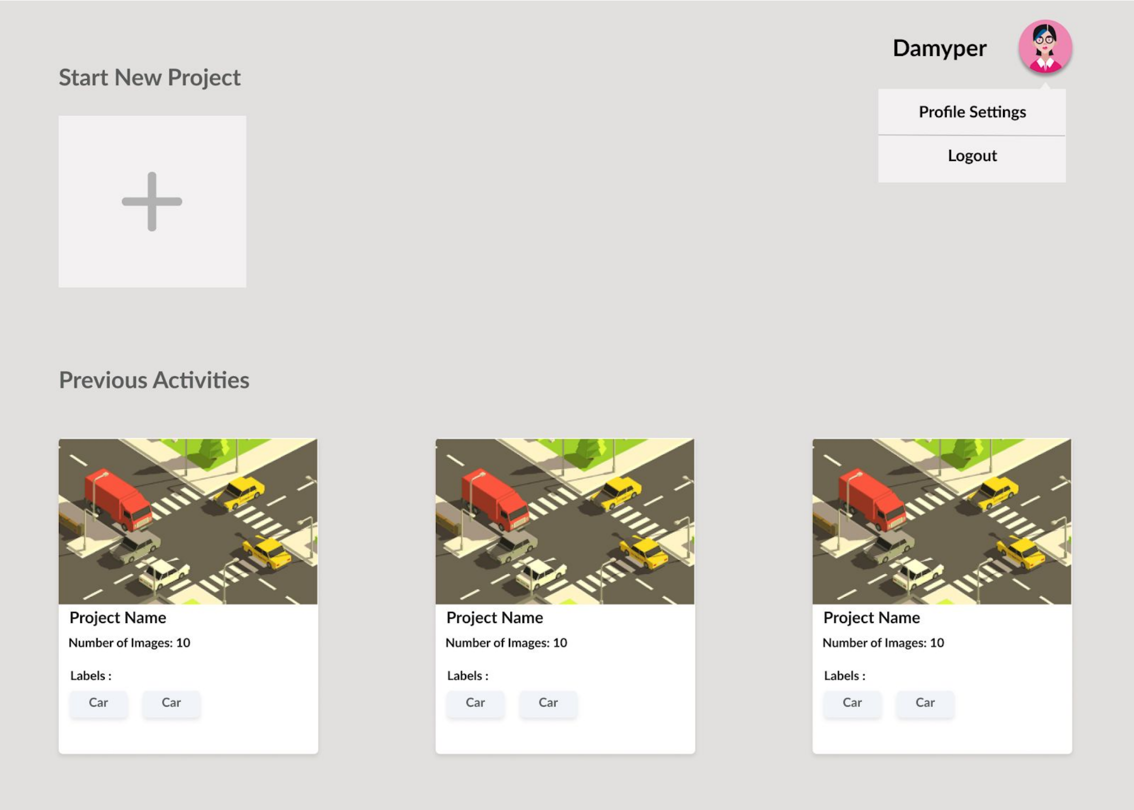
- Implement authentication via Google, Facebook and GitHub (extend for other platforms in future)
    - Passport.js will be used to authenticate the user with Google, Facebook and GitHub.
    - After authentication, the access tokens will be stored for getting the user's private information like email address.

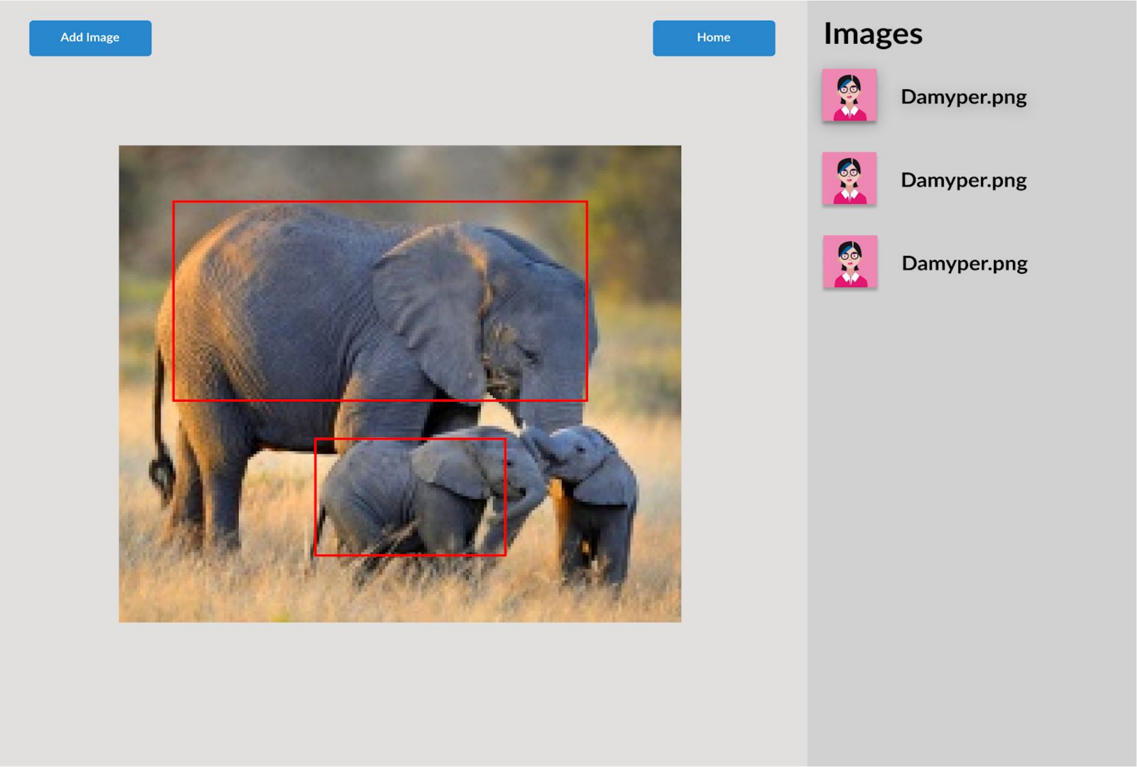
  - These access tokens stored in local storage of the browser for the authenticate check of the user.

2. LabelHub - The dashboard of LabelLab

- A dashboard, on the website, is typically the control panel for a website's content management system. It typically indicates items which require urgent actions at the top of the page, moving into less important statistics at the bottom.
- LabelLab dashboard for now mainly consists of three parts: *Project Creation*, *Navbar* and *Recent Activities or Previous Projects* sections.
  - **Project Creation:** There will be a create new project button or card as they are common website feature because they add visual interest and encourage visitors to click a link or engage with your content.
  - **Navbar:** A navigation bar is a user interface element within a webpage that contains links to other sections of the website. (see Figma)
  - **Recent Activities or Previous Activities:** Recent Activity card which display the image selected, all the labels and view analytics option. The card is a snapshot-like display intended to encourage users to click or tap to view more details.

- All user interface will be developed in React, a javascript library for building user interfaces and Redux as a state management tool. The styling of components will be done in CSS.
- As a feel of what Label Hub components could look like, I designed some quintessential ones. The mockup components and screens have been designed on Figma, which is a tool more suited to the task as compared to a document. I'll be thankful for any feedback or ideas to improve these designs.


Damyper
Start New Project
Profile Settings
Logout
Previous Activities
Project Name
Number of Images: 10
Labels :
Car
Car
Project Name
Number of Images: 10
Labels :
Car
Car
Project Name
Number of Images: 10
Labels :
Car
Car


Add Image
Home
Images
Damyper.png
Damyper.png
Damyper.png

- This Figma document linked here shall be a living document that will be fine-tuned and refined as the components are being built. It will, in the end, be handed over to ScoreLab to share with all designers enabling them to prototype with components as they are supposed to look in real life.

3. Labeler - The image labelling tool

The Labeler is an image labelling tool of Labellab having a very intuitive interface to use and have full good support to increase Labeler productivity. User can create different projects and can upload a number of images in every project and can label them.

For now, this tool has the feature to create a rectangular box around an object but can be further extended to polygons.

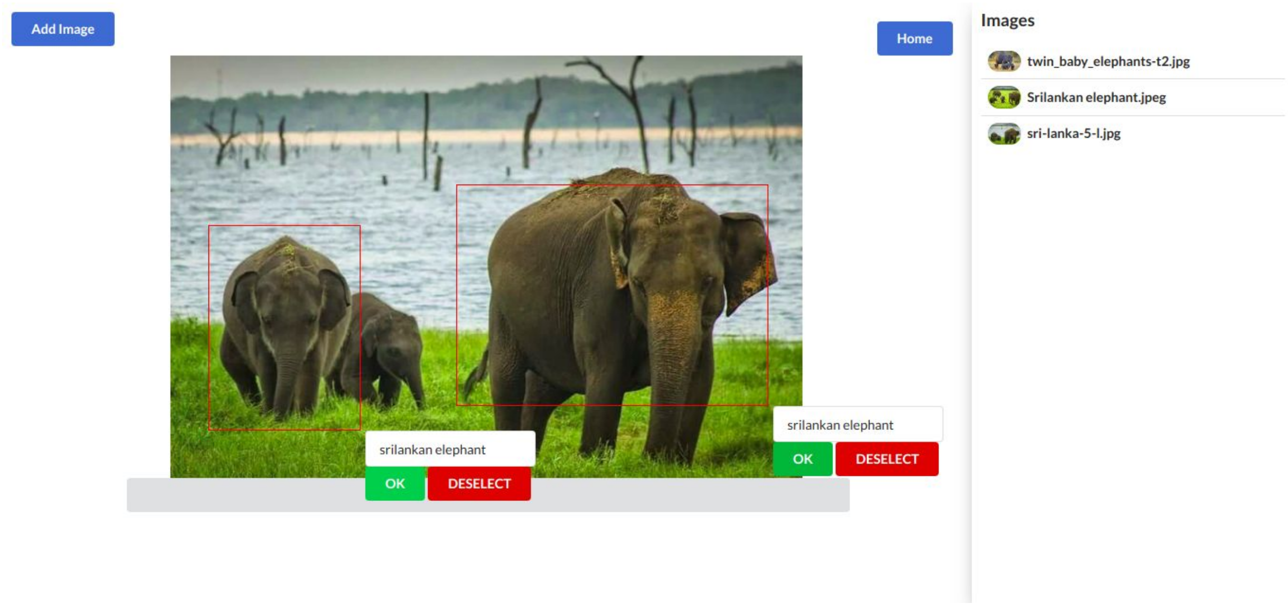
Creating a rectangular box on any object in the image can be implemented using mouse event listeners and some basic functions to find the mouse cursor position and setting it like:

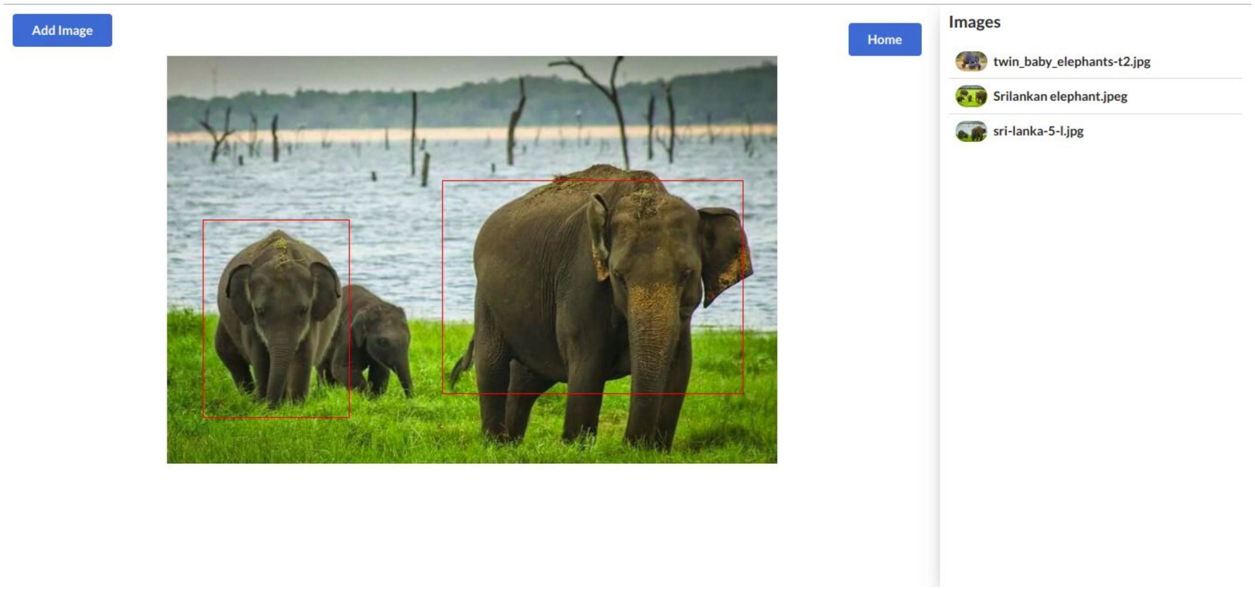
```
initDraw(document.getElementById('canvas'),)

function initDraw (canvas,) {
       canvas.onmousemove = function (e) {
           // ....
       }
       canvas.onclick = function (e) {
           // ....
       }
}
```

```
function setMousePosition(e) {
   var ev = e || window.event; //Moz || IE
   if (ev.pageX) { //Moz
       mouse.x = ev.pageX + window.pageXOffset;
       mouse.y = ev.pageY + window.pageYOffset;
   } else if (ev.clientX) { //IE
       mouse.x = ev.clientX + document.body.scrollLeft;
       mouse.y = ev.clientY + document.body.scrollTop;
   }
};
```

After selection of a number of labels, the user can save the image with the labels to the database. Some images of Labeler from my prototype:

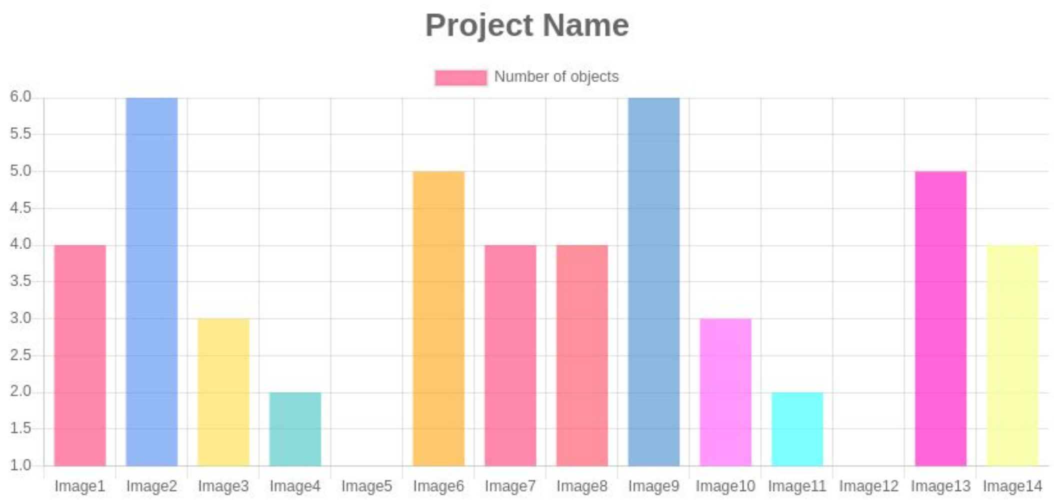




4. Label Analytics

Analytics is the discovery, interpretation, and communication of meaningful patterns in data and the process of applying those patterns towards effective decision making. In LabelLab we can analyze the uploaded data.

After testing the uploaded data on the ML model, we get results on the basis of various image processing parameters.

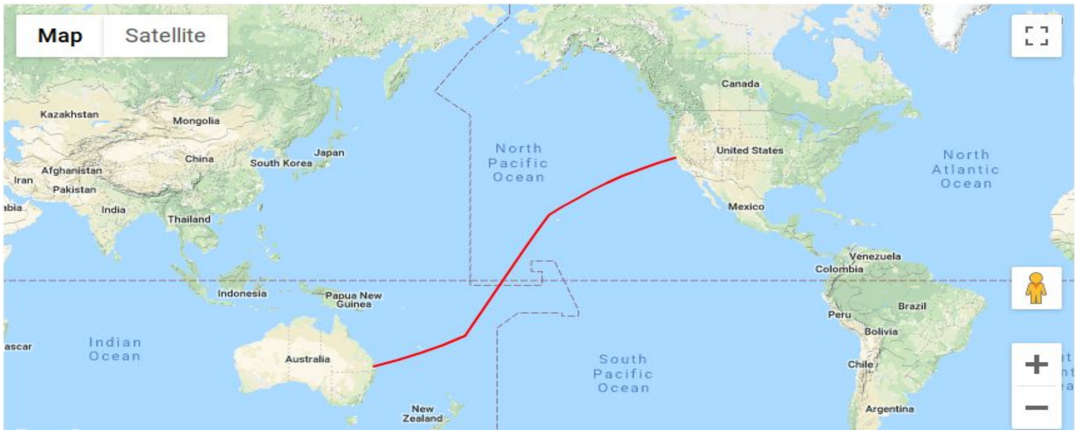


I am supposed to use Chart.js library for drawing different graphs like Pie Chart, Line graph, Bar graph, etc. Integrating react components and redux with chart.js for loading all the data stored in the backend in the chart form.

5. Path Tracking feature

- **Path Tracking feature** will be implemented in the number of images using **ExifTool,** a tool which captures GPS coordinates and stored them as "metadata" which will be embedded in the photo.

- Latitudes and longitudes extracted from the image's metadata will be traced on the map with Polylines feature of google maps. This tool uses HTML canvas and Google maps for tracing a line passing from the given coordinates.



6. Documentation, Dockerization and Continuous Integration of LabelLab

Documentation is a physical description of a system, device, material, or process. I will use Sphinx as a documentation tool.

I am supposed to use Docker container and build an image for this project. Then the system will be a lightweight, self-contained system and guarantees this project will always run the same, regardless of where it's deployed. It will enable easy deployment on any platform, like Google Cloud Platform, AWS, etc. This Dockerised setup is the preferred mode of installation.

Containerisation can be done on Docker and Orchestration on Docker Compose.

Continuous Integration (CI) is a development practice that requires developers to integrate code into a shared repository several times a day. I am supposed to use Travis for continuous integration, for that we have to write a file **.travis.yml** consisting of all the information of app.

Timeline Breakup

Timeframe	Start Date - End Date	Task
Phase 0: Community Bonding	May 6 - May 12	- Interact with the mentors of the project and set up feedback loops. - Get familiarised with the code base. - Explore the technologies to be used in the project.
	May 13 - May 19	- Continue to refine the plans for the project in consultation with the mentors. - Setup the project tracking tooling such as **Trello** or **GitHub** projects.
	May 20 - May 26	- In the case of new ideas developed here will be opened as issues in the library for discussion.

		- Finalize the idea and structure of the project.
Phase 1:	May 27 - June 2	- Set up backend server on Node.js and express.js. - Integrate MongoDB with the server with mongoose as its ODM and create models. - Implement authentication with Google, Facebook, GitHub.
	June 3 - June 9	- Start working on API endpoints to communicate with the database. - **REST API's** of user actions and controls like login, register, creating projects, editing of user profile and projects, image upload, etc. - Simultaneously write tests for them.
	June 10 - June 16	- Initialize the LabelLab client side in React.js and Redux. - Integration of eslint as the linting tool. - Authentication Components (Login/Register) with validations on the form.
	June 17 - June 23	At the end of phase I, the following features will be completed: - Endpoints of all the API's will be completed. - Front End will be completed till the authentication part. - Wrap up the backend and frontend tasks and prepare for first evaluation.
Phase 2:	June 24 - June 30	- Start working on **Label Hub** (dashboard

		of the user) including UI designs. - Edit profile form handling and navbar of LabelLab - Start working on the **Labeler** - the labelling tool of LabelLab
	July 1 - July 7	- **Labeler** tool user interface and mouse event listeners for flexibility of labelling tool. - After selection of an object, form handling for the naming of a label. - Single or multiple image upload with the naming of image and URL. - User guide for using the tool. - Simultaneously testing the uploaded data in the ML model. - Previous activities or recent activities card on dashboard featuring details of the project like the number of images, labels and many more.
	July 8 - July 14	- Analytics and summary of each project data uploaded by the user and for the results of the ML model. - Will be Implemented using chart.js. With chart.js, bar graphs, pie charts, histogram or line charts will be prepared for making analytics more informative and understandable. - Make the statistics view customizable.
	July 15 - July 21	At the end of phase I, the following features will be completed: - **Label Hub**(dashboard of LabelLab) will be mainly designed and developed with some additional features remaining.

		- **Labeler**(image classification tool) will be in a fully functional state. - Finalize the previous work before the evaluation.
Phase 3:	July 22 - July 28	- **Path Tracking feature** will be implemented in the number of images using **ExifTool (**extracts metadata)**.** - Simultaneously testing the uploaded data in ML model.
	July 29 - August 4	- Marking of latitudes and longitudes extracted from the image's metadata on the map with Polylines feature of google maps. - Dockerize the LabelLab (node as well as react) and write a docker-compose file.
	August 5 - August 11	- Integrating build and test cases with **Travis**. - Wiki documentation for deploying the platform in the user's computer. - Write documentation in Sphinx**.** - Fixes bugs and refactor the API's and react components.
	August 12 - August 19	- Polish up the UI and review the documentation. - Prepare for the final evaluation.
Finalize	August 19 - August 26	**Final Evaluation**
Post-GSoC Period	-	Continue to work on the project, finishing items that have been put in the extra milestones sections. Improve the codebase in aspects that will only come to light when integrations

		are getting merged.
Distant Future	-	In the distant future, I would love to work with ScoreLab as they explore more researches and strategies that lead to sustainability. I would also love to mentor new-coming contributors to the project in any way I can.

Reference

1. https://docs.google.com/document/d/1ixt93Z1WlllarTnvI4zTARknNa9_QxQpFLJ1WOHqNhU/edit
2. **ScoreLab Official Website:** http://www.scorelab.org/
3. https://github.com/scorelab/GSoC