Generating Usability Reports from User Inputs and Eye Movements

A PROJECT REPORT

submitted by

Ganesh Sekhar (CHN16CS055)
Shan Eapen Koshy (CHN16CS095)
Sachin Sajan Punoose (CHN16CS)
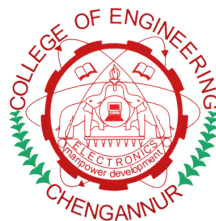**S Hemanth (CHN16CS098)**

to

The APJ Abdul Kalam Technological University
in partial fulfillment of the requirements for the award of the Degree

of

Bachelor of Technology

In

Computer Science and Engineering

Department of Computer Science and Engineering

College of Engineering, Chengannur, Kerala -689121

July 2020

DECLARATION

We undersigned hereby declare that the project report "Generating Usability Reports from User Inputs and Eye Movements", submitted for partial fulfillment of the requirements for the award of degree of Bachelor of Technology of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by us under supervision of Ms.Angel, Assistant Professor. This submission represents our ideas in our own words and where ideas or words of others have been included, We have adequately and accurately cited and referenced the original sources. We also declare that we have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. We understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Place: CHENGANNUR   Ganesh Sekhar

Date:2020/07/16   Shan Eapen Koshy
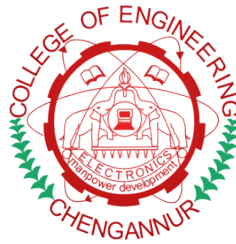
                                       Sachin Sajan Punnoose

                                       S Hemanth

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

COLLEGE OF ENGINEERING,CHENGANNUR

CERTIFICATE

This is to certify that the report entitled **"Generating Usability Reports from User Inputs and Eye Movements"** submitted by **Ganesh Sekhar**, **Shan Eapen Koshy**, **Sachin Sajan Punnoose**, **S Hemanth** to the APJ Abdul Kalam Technological University in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Department of Computer Science and Engineering, College of Engineering, Chengannur, Kerala -689121 is a bonafide record of the project work carried out by them under my/our guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose.

Ms. Angel Thankam Thomas   **Ms.Shiny B**
Assistant Professor   Assistant Professor
Project Guide   Project Co-ordinator

Dr. Smitha Dharan
Professor and Head

ACKNOWLEDGEMENT

ABSTRACT

Usability testing is a technique used to evaluate a product by testing it on users. It is an important factor in marketing a product since it gives a complete structure of how the users use the product.

After understanding how real users interact with your product, you can improve the product based on the results. The primary purpose of a usability test is to improve it's designed so as to make it more user-friendly.

The proposed system uses eye detection to locate the positions on the screen where the user pays more attention and a heat map is generated from it. This testing is done for different age groups and a final report listing all the findings (positives and negatives) is generated. Positive findings will help the team to know that they're on the right track and the negative findings provide proposals to solve them

Contents

List of Figures

INTRODUCTION

Usability testing is the practice of testing how easy a design is to use on a group of representative users. It usually involves observing users as they attempt to complete tasks and can be done for different types of designs, from user interfaces to physical products. It is often conducted repeatedly, from early development until a product's release. The main benefit and purpose of usability testing is to identify usability problems with a design as early as possible, so they can be fixed before the design is implemented or mass produced. As such, usability testing is often conducted on prototypes rather than finished products, with different levels of fidelity (i.e., detail and finish) depending on the development phase. Prototypes tend to be more primitive, low-fidelity versions (e.g., paper sketches) during early development, and then take the form of more detailed, high-fidelity versions (e.g., interactive digital mock-ups) closer to release.

In a typical usability test, a test moderator gives test participants a series of tasks that they must perform with the design. The tasks represent actions that an end user would typically carry out with the finished product. During the test, the moderator observes each participant's actions, often also recording the test session on video. After analyzing the results of a usability test, the moderator reports on several points of interest that arose—these include issues such as the aspects of the design that caused problems and the severity of these problems, as well as places in the design that the participants particularly liked. Recognizing this potential to highlight difficulties and strong points in a design's early versions is a vital part of a designer's thought process. The broader the testing and the greater the number of matters raised, the stronger the likelihood that designers can craft more successful products.

Traditional usability methods and performance measurements might indicate that there's an efficiency issue, but often do not answer why or how to fix it. Eye tracking uniquely provides information about tasks that are not articulated by participants and that might otherwise pass unobserved by the researcher. It captures natural, unbiased user behavior and produces objective data to allow effective recommendations to be made. Eye tracking is a flexible technique that works with a variety of research methods, including observations, interviews, and retrospective think aloud (RTA). Our approach combines eye tracking with several other data points such as cursor movements, mouse clicks, hover duration and more to score the UI elements present in the screen to generate an interactive report.

CHAPTER 2

PROBLEM FORMULATION

Usability testing is crucial in a software development life cycle as it provides more insights on how a user uses the product. Typically, a UX researcher summons the tester to his/her office and has to manually observe and analyze the user to validate the designs. But this traditional usability testing approach takes huge amount of time, money and workforce. This in turn increases the software development time which causes late delivery of the product. Different usability testing metrics that UX designers uses are:

- Focus points of the users on the screen

- Time taken by the user to find the target action/data he/she was looking for

- Session duration

To find the focus points, the UX researcher asks the tester to move a pointer across the screen which is prone to errors. The other metrics are also manually recorded which are prone to errors. To overcome this we propose a novel method to automate usability testing that accounts various other metrics including eye tracking.

<div align="center">

CHAPTER 3

LITERATURE REVIEW

</div>

3.1 Remote Usability Testing Using Eyetracking[1]

In this paper, a low cost method of using eyetracking to perform remote usability tests on users is described. Standard web camera with freeware software is used for this experiment and the experiment concludes that it is not perfect, but it lays a foundation for futher development. Also, the technique described in the paper is not realtime and manual post processing is required for generating reports.

3.2 Eye Tracking in User Experience Testing: How to Make the Most of It[2]

This paper introduces eye tracking as a user experience testing tool. It focuses on how to design and conduct studies involving eye tracking, so that eye movement data can effectively supplement data obtained through more conventional methods. Using examples from actual studies, we share lessons learned and provide advice on how to avoid common mistakes.

3.3 Webgazer: Scalable Webcam Eye Tracking Using User Interactions[3]

WebGazer, an online eye tracker that uses common webcams already present in laptops and mobile devices to infer the eye-gaze locations of web visitors on a page in real time is introduced in this paper. The eye tracking model self-calibrates by watching web visitors interact with the web page and trains a mapping between features of the eye and positions on the screen. This approach aims to provide a natural experience to everyday users that is not restricted to laboratories and highly controlled user studies. WebGazer has two key components: a pupil detector that can be combined with any eye detection library, and a gaze estimator using regression analysis informed by user interactions.The findings show that WebGazer can learn from user interactions and that its accuracy is sufficient for approximating the user's gaze.

3.4 Eye Tracking in User Experience Design[4]

CHAPTER 4

RELATED WORKS

4.1 Manual User Testing

Manual usability testing refers to evaluating a product or service by testing it with representative users. Typically, during a live test, participants will try to complete typical tasks while observers watch, listen and takes notes. The goal is to identify any usability problems, collect qualitative and quantitative data and determine the participant's satisfaction with the product.

Manual testing is an activity, where testing of an application is done by human. Software tester (but it can also be a business user on the client side) executes tests based on the defines test cases ensuring that the application is working properly and meets defined requirements.

But manual testing has many disadvantages; Some tests are difficult or almost not testable manually. E.g. performance tests. Less reliable testing results as tests are performed by human. Especially during peaks and under pressure. Could be a time-consuming activity, especially when re-executing regression tests multiple times. Running manual regression tests may lead to team de-motivation and potentially to professional blindness.

4.2 Usertesting.com

UserTesting.com provides web, desktop and mobile app testing in the market. The company gives marketers, product managers and UX designers, on-demand access to users in their target audience, who deliver audio, video and written feedback on websites or apps in less than one hour. UserTesting.com offers usability testing with advanced targeting, expanded recruiting, live intercepts, moderated testing, quantitative metrics, and research and reporting services. Their usability report consisting of audio and video recordings of rapid customer testing, via video conferencing medium. Its very much similar to manual testing. The only difference is that it enables remote users to engage in the testing. So the disadvantages of manual testing affects this model of testing as well.

4.3 Hotjar

Hotjar is a tool that reveals the online behavior and voice of your users. By combining both Analysis and Feedback tools, Hotjar gives the 'big picture' of how to improve your site's user experience and performance/conversion rates. The Analysis tools allows one to measure and observe user behavior (see what users do) while the Feedback tools enable you to "hear" what your users have to say (the Voice of User). But hotjar does not provide us with eye gaze spots or the focus points of the user. The only factor driving the usability testing report by hotjar is mouse pointer and its clicks.

4.4 UXReality

CHAPTER 5

PROPOSED SYSTEM

5.1 Overview

In this proposed system, a user can submit a URL of the website to be analyzed. The system then generates a unique tracking code for this website which can be manually inserted into the website to be tested. Testers can access this URL and interact with the website normally while we collect the tester's eye coordinates that we obtained through webgazer.js. Basic demographic of the tester such as age and gender are also collected for categorization and report generation. The collected data is then stored in the server. The testing details can be reviewed from the admin's dashboard. Several features such as timeline, demographic filtering, heatmap,AOI, etc, are provided for easily analyzing the data.
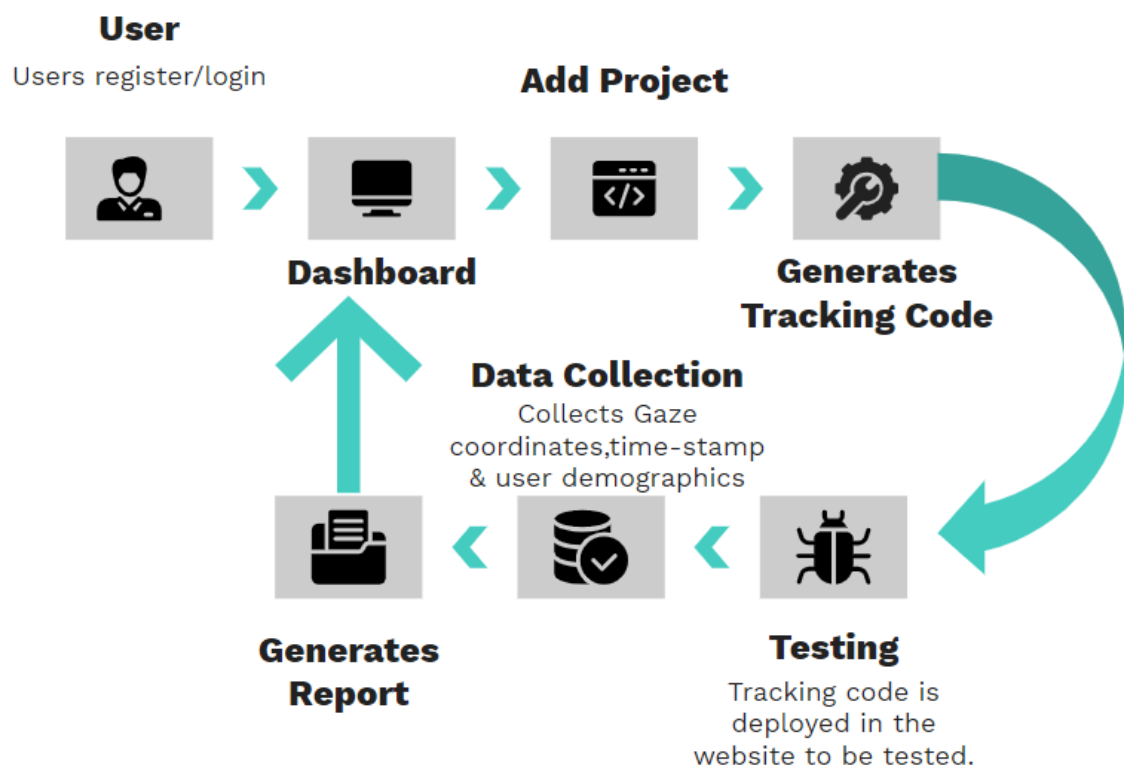

Figure 5.1.1: Control Flow Diagram

The system mainly consist of 2 parts:

- Client Side Script

- Backend & Dashboard

5.2   Website Tracking Script

The client side script is a javascript file that performs the eye tracking calibration, eye track-ing and user data collection. The script is also responsible to preload all the necessary dependen-cies including webgazer.js, jQuery and many more. To avoid the users from having to manually include each dependency and to prevent a dependency deadlock scenario, all the dependencies are packaged into a single script using webpack.

It mainly consist of the following modules:

5.2.1   Tracking Code

The tracking code is the only piece of code that a website manager needs to add to his/her site. A unique tracking code is generated for every project and the code is inserted into the ***head*** tag of all the required webpages. The tracking code stores the unique variables of a project such as the project ID and other paramaters which can then be accessed from the tracking script.

Tracking Code

Add the following code just below the closing head tag.

```HTML
<script>
var UXQ = UXQ || {};
UXQ['projectID'] = "UX-12247";
UXQ['track'] = ['clicks'];
</script>
<script defer src="https://userxp.ga/userXP.js"></script>
```

Figure 5.2.1: Tracking Script

5.2.2   Calibration & Eye Tracking

When the website to be tested is loaded it's content is replaced by the calibration page. During the calibration process the system learns how a tester's eyes move when they are looking at certain parts of the screen. WebGazer, a self callibrated eye-tracking library is used for obtaining the eye-gaze locations of varoius testers on the website in real time.

In the initial phase of calibration the eye tracking model is self callibrated by matching the pupil positions and eye features with screen locations during user interactions. Upon completion the calibration page hides away revealing the website which the users can interact normally for the specified duration which is defaulted to 30 seconds. On hitting the specified time interval, a form appears and prompts the tester to submit his/her basic details such as age and gender. The

tracking results are also submmited along with the form.

```
    [ 539, 456, 1564, 0, 771, 426 ],
    [ 539, 456, 1587, 0, 771, 426 ],
    [ 539, 456, 1605, 0, 771, 426 ],
    [ 539, 456, 1624, 0, 771, 426 ],
    [ 539, 456, 1638, 0, 771, 426 ],
    [ 538, 400, 1654, 0, 771, 426 ],
    [ 538, 400, 1670, 0, 771, 426 ],
    ... 492 more items
  ],
  _id: 5eeb3a4b642ca885a0464327,
  projectID: 'UX-73636',
  date: '1592474177163',
  page: 'http://localhost:8080/',
  browserWidth: 1536,
  browserHeight: 864,
  tester: { name: 'hemanth', age: 22, gender: 'M' },
  __v: 0
}
```

Figure 5.2.2: Tracking Data

The above figure shows a snippet of the data being posted to the server after the test. The submited data consist the following data:

- Project ID: A unique ID to identify each project.

- URL: URL of the website to be tested.

- Browser Dimentions: It consist of the browser width and height in pixels.

- User Demographics: Details such as name, age and gender of the tester.

- Gaze-Recording: It consist of 5 attributes-Eye Coordinates(both x & y coordinates), Timestamp, Scroll Offset and Mouse Coordinates(both x & y coordinates) respectively.

5.3   Dashboard

It's an interactive website made using HTML,CSS & JS which acts as platform for the users for adding,managing and the deletion of their projects. Each of the projects page is equipped with various details such as tracking code, user demographic chart and the listing of all the testers along with the report.

5.3.1   SighUp and Login

The website manager can register in our platform by Signing Up the form directly or by using Sign Up using Google option.

In Sign Up directly the user can create an account by filling the form. The password is encrypted using MD5 encryption algorithm and the user details are stored into the database. Users can also signup using their Google accounts, where basic details like name, email and image will be retrieved from Google and will be stored in our database.

5.3.2   Project Management

Dashboard also acts as a platform where the users can create, update and delete projects. Once the user logs into the dashboard all his existing projects will be displayed. Method for adding new projects is also displayed.

Project Creation

The prompt for creating a new project is shown below. In this the user have to enter the name of the project as required, the URL of the website to be tested and the maximum number of testers required for this project. Once the project is created, the above details are added into the database.
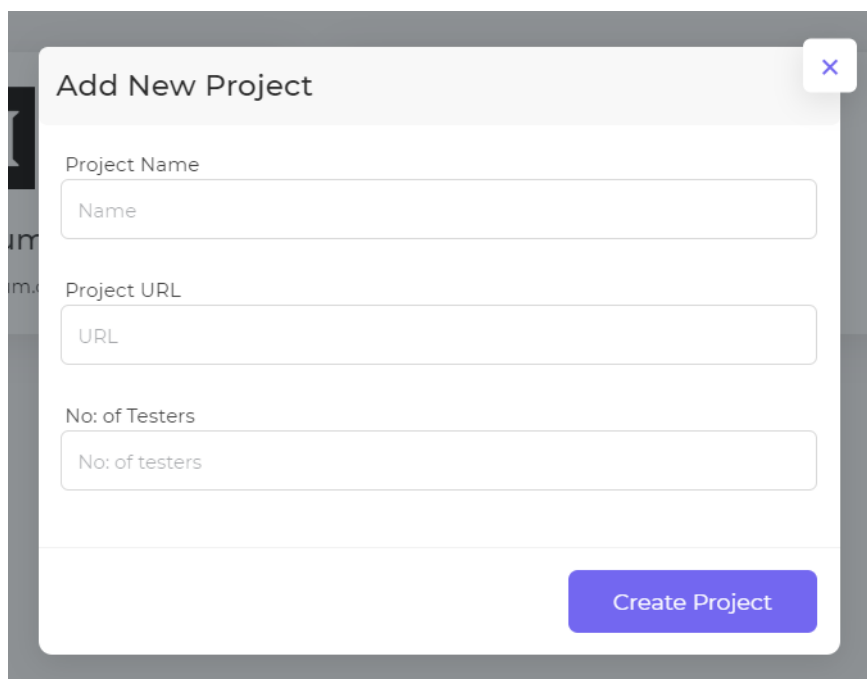

Figure 5.3.1: Create new project prompt
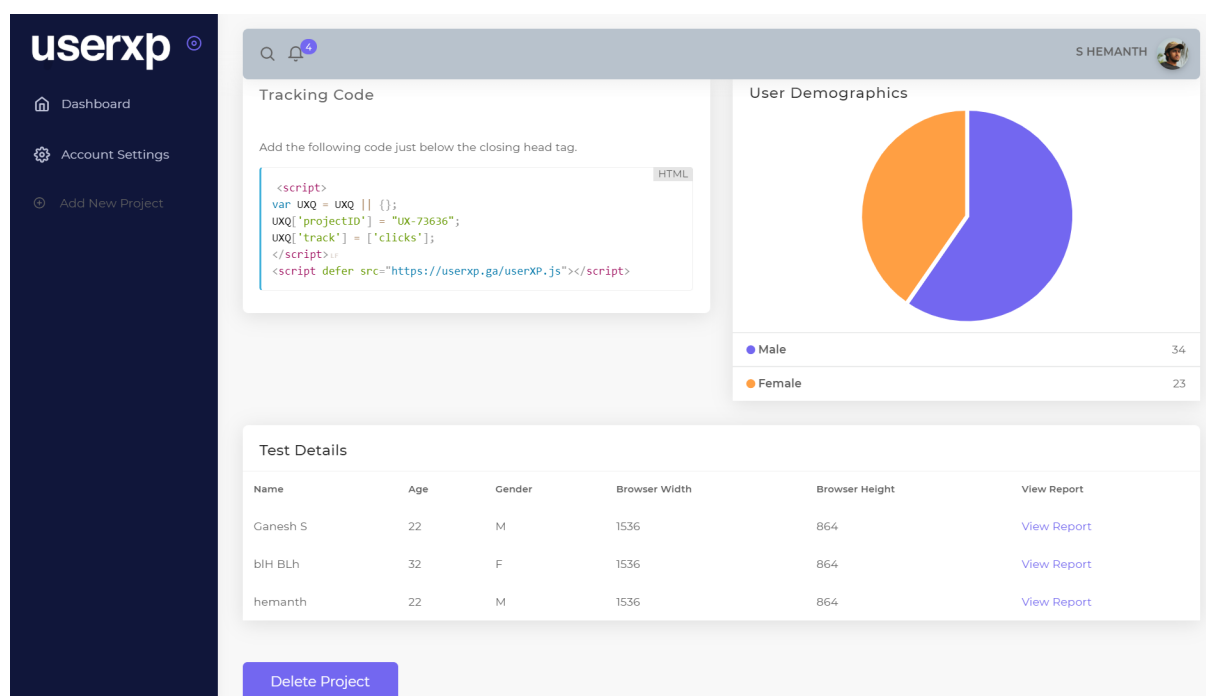
Project Details

Figure 5.3.2: Project Page

Each project page contains the unique tracking code for that project which is to be inserted into the website to be tested. A pie-chart displaying the details of the tester demographics and a table showing the details of all the testers for this project along with their individual report. An option to delete the project is also available. Once the project is deleted the project data along with the tester's data is removed from the database.

In order to avoid the accidental deletion of the project, the user is asked to enter the Project-ID to conform the deletion process.
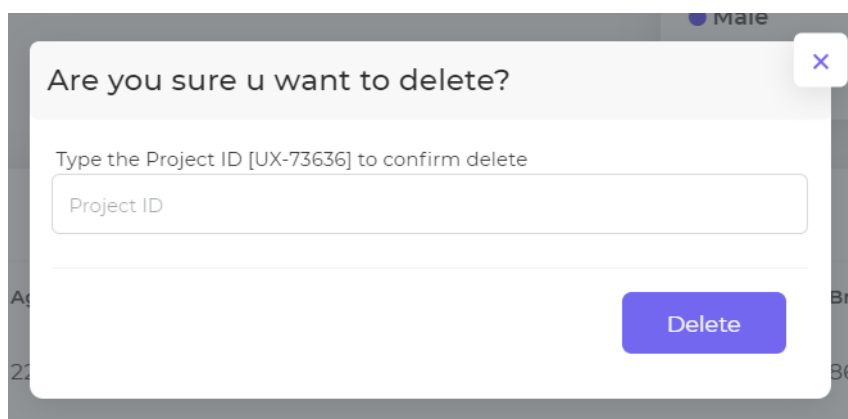

Figure 5.3.3: Delete Project Prompt

5.3.3 Usability Report and it's Features

A usability report is generated from the data obtained from each testes of a project which provides an insight on how various testers interacts with the website and infer the necessary changes to be made so as to improve the user experience.

Various features of the report are:

Heat map & Gaze Opacity Map

A heatmap is a visualization that uses different colors to highlight the regions viewed by the tester. Heat maps are color coded: red is typically used to indicate the most focused region and green the least, with varying levels in between. An area with no color on a heat map signifies that the participants may not have focused on the area.
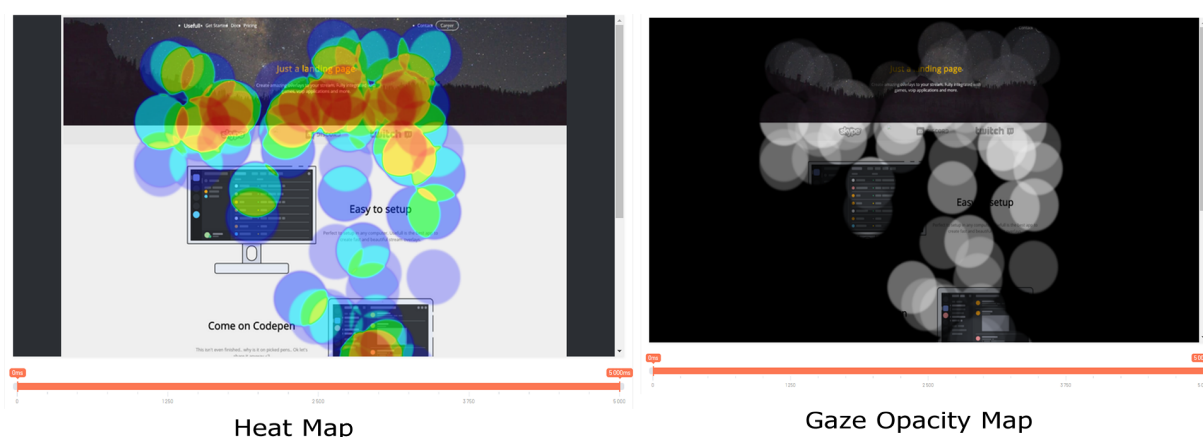

Heat Map   Gaze Opacity Map

Figure 5.3.4: Heat Map & Gaze Opacity Map

The Gaze Opacity map is an inverse map of compiled fixation data. The opposite of a heat map, it is best used to highlight areas of user interaction on the website.The white region in this map marks the region where the tester had focused and the black region corresponds to less interaced area. It is also known as a Spotlight map.

Area of Intrest(AOI)

AOIs are specific areas or content of the user interface that interests the UX team. The user can select any specific portion on the website by drawing a square box in the window and can obtaine the details on how much time the tester had spend in that region.
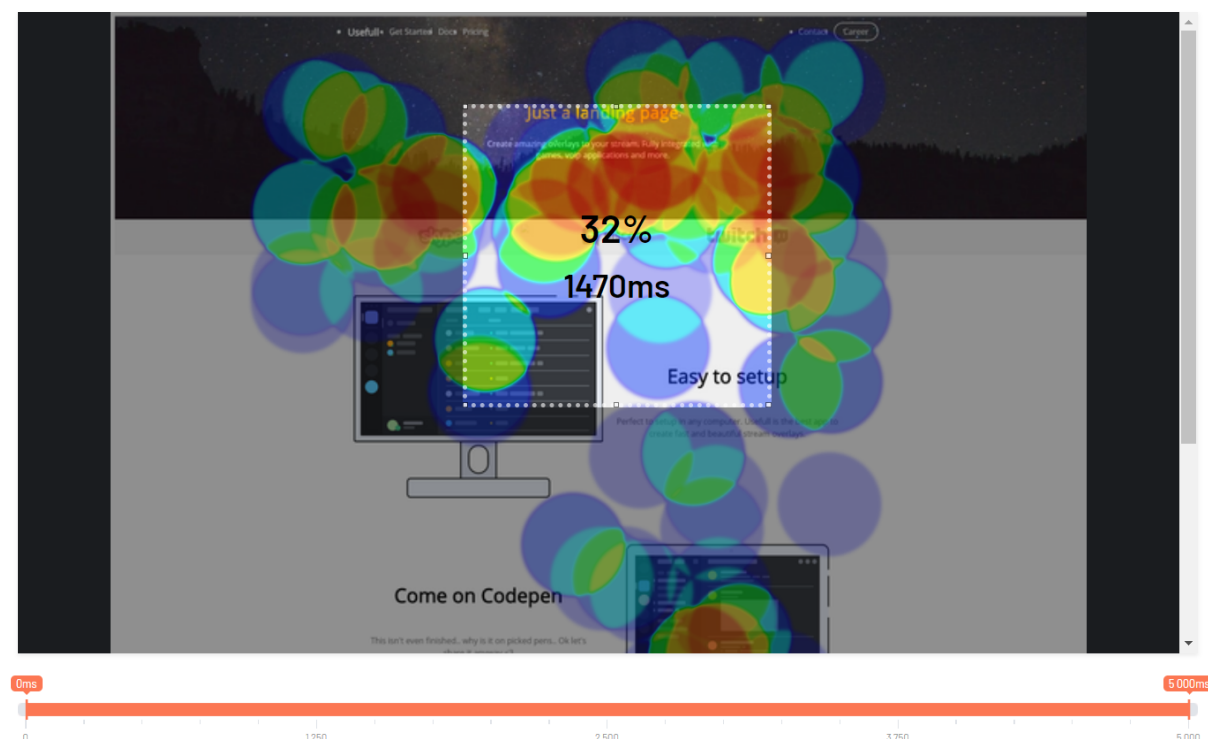
Figure 5.3.5: Area of Intrest(AOI)

In the above figure,the square portion marks the Area of Intrest and the data present shows that the user had spent 32% of the total time in this region and spend about 1470ms here.

5.4   Backend

The backend of our system is powered by NodeJS with Express and peforms the tasks such as database management, routing, templating, authentication and it also provides API endpoints for the frontend.

NodeJS is a JavaScript runtime built on Chrome's V8 JavaScript engine. It was used due to its high performance and the due to the fact that client side script is written in javascript.

EJS or Embedded Javascript Templating is a simple templating engine used by NodeJS that lets you generate HTML markup with plain JavaScript.It can inject data into HTML template at the server side and produce the final HTML. It is used because of it's fast compilation and rendering property.

MongoDB is leading NoSQl, cross-platform, document oriented database that provides, high performance, high availability, and easy scalability. MongoDB works on concept of collection and document. The data is stored in BSON(Binary JSON) format in the form of key-value pair.

5.4.1   Database

MongoDB was used as database due to the nature of data being dealt with. MongoDB uses JSON-like documents with optional schemas. The entire project runs on a single document with 3 collections.
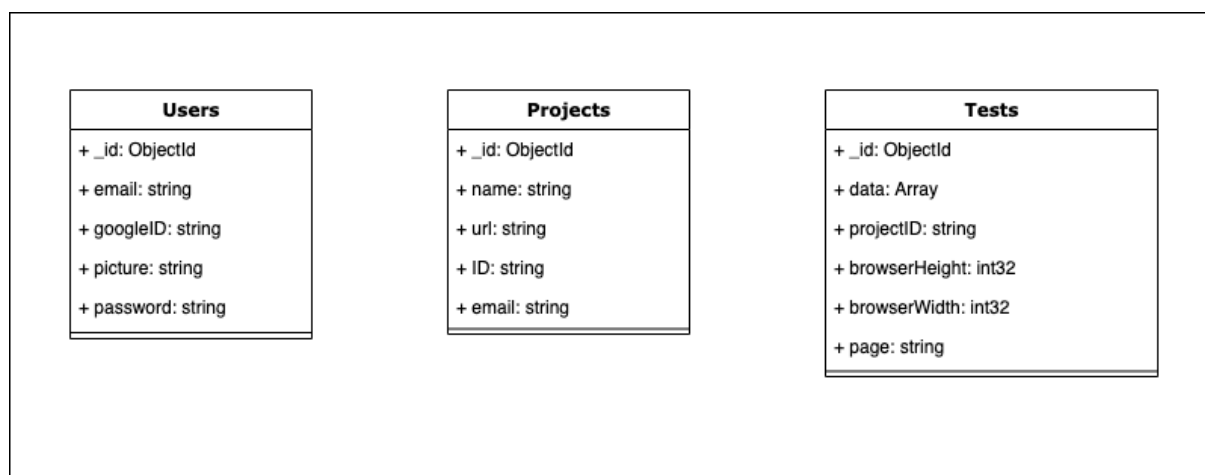
Figure 5.4.1: MongoDB Collections

The figure shows the basic schema for the 3 collections but since MongoDB requires no schema we can easily add new entries to each document as a key-value pair.

5.4.2 Authentication, Routing, Templating & Other Tasks

One of the first task of the backend is to authenticate users and we provide two methods for doing that. Users can signup using Google SSO or with an email and password. Authentication is handled by passportJS, a nodeJS library and users are authenticated with a cookie based approach.
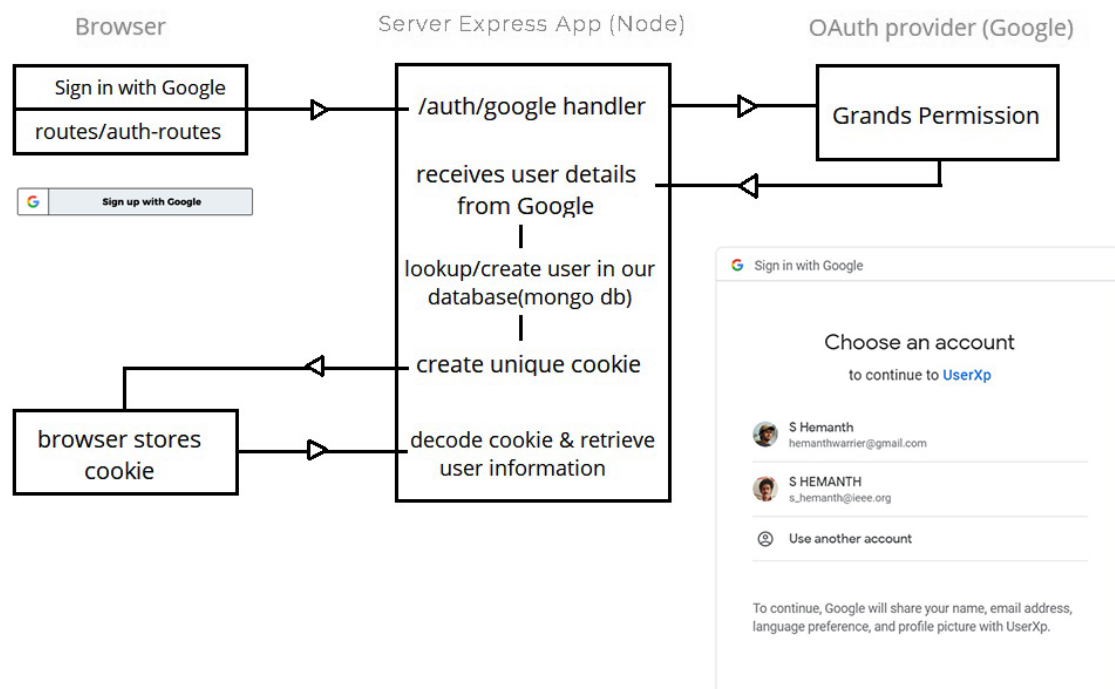

Figure 5.4.2: Google SSO Authentication

Routes are defined using Express which is the goto framework for the purpose. It allows us

to recieve and respond to different types of http requests like GET, POST or DELETE.

Next, the backend generates the requested dashboard pages using EJS templating and is served as plain HTML files.

5.4.3 API Endpoints

The following APIs allow the frontend to interact with backend for CRUD operations.

- **/api/v1/submit** This endpoint is used to recieve the form data from the website and it is available publically. To avoid misuse or false submissions, each request is verfied by crossmatching the origin URL and the URL in the database for the given project ID.

- **/api/v1/newProject** This endpoint should originate from an authenticated user and it creates a new DB entry with the given project details.

- **/api/v1/delete/:id** The delete endpoint takes an ID of the project which is to be deleted. The backend verifies that the request is coming from an authenticated user and also if the request originated from the corresponding project page. This ensures that other projects are not accidently or intentionally deleted by bruteforcing the id in the endpoint.

CHAPTER 6

CONCLUSION & FUTURE SCOPE

At present, existing Usability/Testing methods for web based platforms are quite expensive and requires a considerable amount of resources including man-power and time. Thus to overcome these challenges, the proposed system uses eye-tracking, cursor-movement and mouse-clicks to evaluate the positions on the screen where the user pays more attention and a score for each UI element is assigned from the resulting heat-map. This testing is done for different age groups a final analysis report is generated. The UX team in turn can use this report to identify what they have done right & wrong and hence improve their design.The default behavior of EJS is that it looks into the 'views' folder for the templates to render. So all the webpages required is to be stored in the views folder in ejs format.

REFERENCES

[1] Chynal, Piotr & Szymański, Jerzy (2011). *Remote Usability Testing Using Eyetracking.*
`https://www.researchgate.net/publication/221053998_Remote_`
`Usability_Testing_Using_Eyetracking`

[2] Agnieszka Bojko (2015). *Eye Tracking in User Experience Testing: How to Make the Most of It*
`https://www.researchgate.net/publication/266161907_Eye_`
`Tracking_in_User_Experience_Testing_How_to_Make_the_Most_`
`of_It`

[3] Papoutsaki, Alexandra & Sangkloy, Patsorn & Laskey, James & Daskalova, Nediyana & Huang, Jeff & Hays, James. (2016). *WebGazer: Scalable Webcam Eye Tracking Using User Interactions.*
`https://jeffhuang.com/Final_WebGazer_IJCAI16.pdf`

[4] Jennifer Romano Bergstrom & Andrew Jonathan Schall *Eye Tracking in User Experience Design.*
`https://books.google.co.in/books?id=5Hp0AgAAQBAJ&lpg=PP1&`
`ots=2Zh1X9Y9u0&dq=eye%20tracking%20in%20user%20experience%`
`20design&lr&pg=PP1#v=onepage&q&f=false`

[5] Kiril Alexiev, Teodor Toshkov and Peter Dojnow. 2019. Accuracy and Precision of eye tracker by head movement compensation and calibration. *20th International Conference on Computer Systems and Technologies (CompSysTech'19)*, Jun 21-22, 2019, Ruse, Bulgaria, 8 pages.
`https://doi.org/10.1145/3345252.3345278.`