VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

TRIM - The URL Shortener

PROJECT REPORT

Internet and Web Programming (CSE3002)

Slot: B1

Submitted in partial fulfilment for the award of the degree of
B. Tech in Computer Science & Engineering.

By

NAME	REG.NO
1. ISHAAN OHRI	18BCE0265
2. SHUBHAM SRIVASTAVA	18BCE2150
3. MITUSHI RAJ	18BCE2201

Under the guidance of
DR. JAYAKUMAR K

November 2020

DECLARATION

We hereby declare that the thesis entitled **"TRIM - The URL Shortener"** submitted by us, for the award of the degree of B.Tech. Computer Science and Engineering, is a record of bonafide work carried out by us under the supervision of DR. JAYAKUMAR K.

We further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place: Vellore
Date: 1-11-20 Signature of the Candidate

CERTIFICATE

This is to certify that the thesis entitled **"TRIM - The URL Shortener"** submitted by **Ishaan Ohri (18BCE0265), Shubham Srivastava (18BCE2150) and Mitushi Raj (18BCE2201)** for the award of the degree of B.Tech. Computer Science and Engineering, is a record of bonafide work carried out by them under our supervision.

The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.
The Project report fulfils the requirements and regulations of
VIT and in my opinion meets the necessary standards for submission.

Signature of the Guide **Signature of HOD**

ACKNOWLEDGEMENT

It is our pleasure to express with deep sense of gratitude to **DR. JAYAKUMAR K**, Associate Professor Grade 1, School of Computer Science and Engineering, Vellore Institute of Technology, for his constant guidance, continual encouragement, and understanding; more than all, he taught us patience in our endeavour. Our association with him is not confined to academics only, but it is a great opportunity on our part to work with an intellectual and expert in the field of Internet and Web Programming.

We would like to express our gratitude to **G VISWANATHAN (Chancellor), SEKAR VISWANATHAN (Vice President), DR. ANAND A. SAMUEL (Vice-Chancellor), S NARAYANAN (Pro-Vice Chancellor)**, for providing an environment to work in and for his inspiration during the tenure of the course.

In jubilant mood we express ingeniously our whole-hearted thanks to all teaching staff and members working as limbs of our university for their not-self-centred enthusiasm coupled with timely encouragements showered on me with zeal, which prompted the acquirement of the requisite knowledge to finalize our course study successfully. We would like to thank our parents for their support.

It is indeed a pleasure to thank our friends who persuaded and encouraged us to take up and complete this task. At last but not least, we express our gratitude and appreciation to all those who have helped me directly or indirectly toward the successful completion of this project.

Place: Vellore
Date: 1-11-20 Name of the Student

TABLE OF CONTENTS

ABSTRACT **6**

INTRODUCTION **7**

PROBLEM STATEMENT 7
TECHNICAL SPECIFICATION 7

EXISTING SYSTEM PROBLEMS **8**

PROPOSED SYSTEM DESIGN **9**

MODULE DESCRIPTION 9
1. LOGIN PAGE: 9
2. SIGNUP PAGE: 10
3. TRIM PAGE: 10
4. ABOUT US PAGE: 10
5. FAQ PAGE: 10
6. STATS PAGE: 10
7. LINKS PAGE: 10
DATA FLOW DIAGRAMS 11
1. GENERATING SHORT URLS 11
2. ACCESSING THE SHORTENED URLS: 11
3. USE CASE DIAGRAMS 11
4. CLASS DIAGRAM 12
5. DATABASE DIAGRAM: 13
CODE SNIPPETS 14
shortener.ts 14
signup.html 16
signup.js 18
login.css 18

RESULTS **20**

USER DETAILS: 21
URL DETAILS: 21
FRONTEND MODULES: 22
Module 1: Login page 23
Module 2: Sign up Page 23
Module 3: Trim Page: 23
Module 4: About Us Page: 24
Module 5: FAQ Page: 24
Module 6: Stats Page 25
Module 7: Links Page 25

CONCLUSION **26**

REFERENCES **27**

ABSTRACT

URL shortening is a technique on the World Wide Web in which a Uniform Resource Locator (URL) may be made substantially shorter and still direct to the required page. This is achieved by using a redirect which links to the web page that has a long URL. For example, the URL "https://example.com/assets/category_B/subcategory_C/Foo/" can be shortened to "https://xmpl.com/Foo. Often the redirect domain name is shorter than the original one.

A friendly URL may be desired for messaging technologies that limit the number of characters in a message (for example SMS), for reducing the amount of typing required if the reader is copying a URL from a print source, for making it easier for a person to remember, or for the intention of a permalink.

Many such services currently exist, however a majority of them are either paid, have incorporated a subscription system without which custom (and more memorable URLs) can't be set up or non-user-friendly interfaces. Besides this the few outliers that exist, have some dated caching and redirecting methods involved, causing long redirection times and occasional deprecation of shortened URLs after a while.

Analysing these, it's clear that there is a need for a more convenient URL trimming service for the public that can overcome these issues.

INTRODUCTION

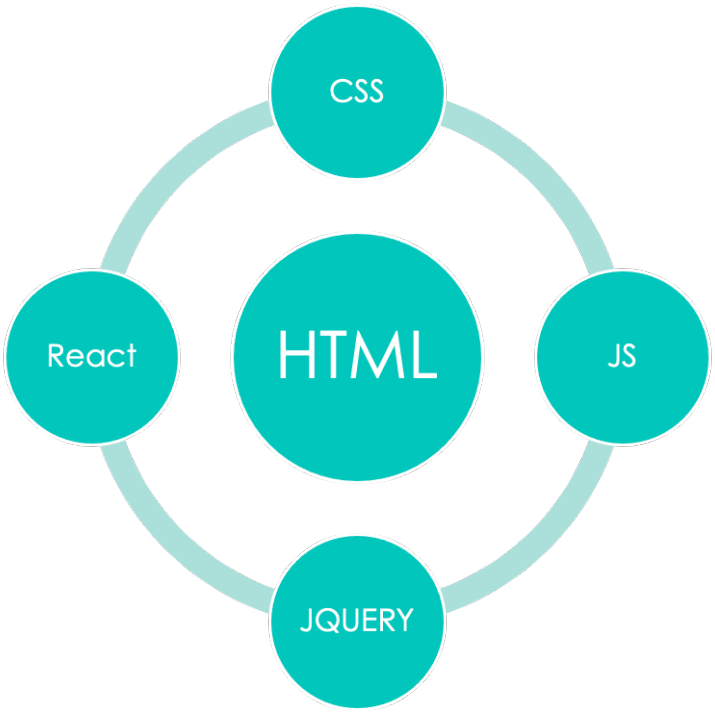
PROBLEM STATEMENT

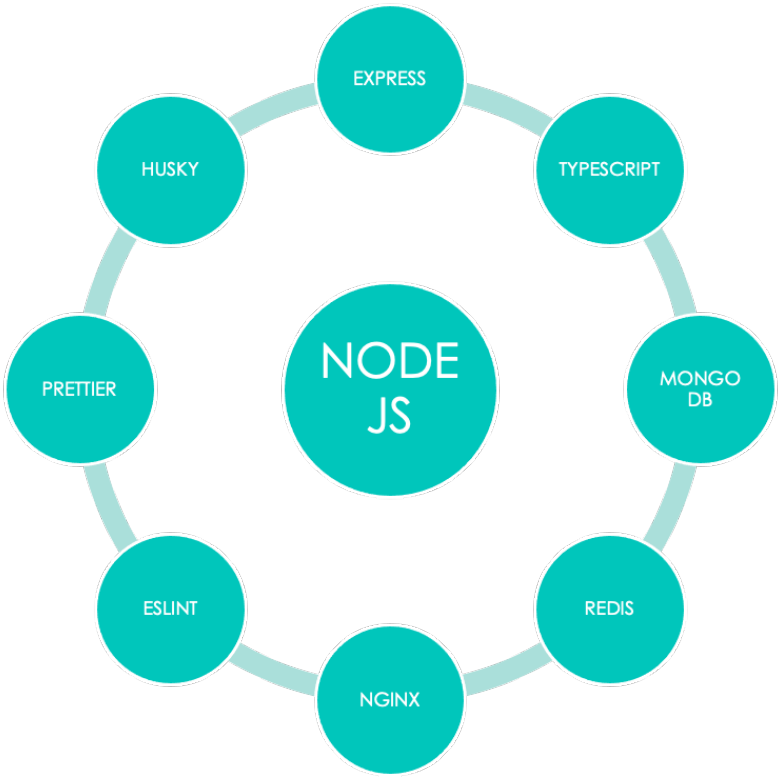
We were inspired to make a URL trimming service that:

- Is fully free of cost
- Allows custom URLs
- Gives a detailed account of statistics and usage information for SEO purposes.
- Minimalistic, aesthetic and convenient UI
- Faster redirection than existing methods by using improved caching methods.
- Accessible on all kinds of devices

TECHNICAL SPECIFICATION

TRIM uses tech stacks like ***HTML, CSS, JS, JQuery and React*** for the frontend. The backend is a custom build backend on ***Node.js***. It used ***Express*** framework for the design of API endpoints. It makes use of ***TypeScript*** and ***JavaScript*** as the main languages for its design. The backend uses packages like ***ESLint and Prettier*** as analysis tools for identifying problematic patterns found in JavaScript code and as code formatters. It uses ***Husky*** to avoid bad commits on GitHub and to ensure correct code patterns. The shortened URLs are stored on ***MongoDB*** and it uses ***Redis*** as a caching service, in order to ensure faster redirects. ***Nginx*** is used as a load balancer and as a reverse proxy. The entire TRIM service is deployed over a ***Virtual Machine on GCP*** and can be accessed via the public URL *https://iwp-project.ml* . The domain also has an ***SSL certificate*** issued from Let's Encrypt authority. We also make use of ***GitHub Actions*** for Continuous Integration and Continuous Deployment.


CSS
React
HTML
JS
JQUERY


EXPRESS
HUSKY
TYPESCRIPT
PRETTIER
NODE JS
MONGO DB
ESLINT
REDIS
NGINX

EXISTING SYSTEM PROBLEMS

Although there exist a lot of URL shorteners out there on the web, yet each one of them has one drawback or the other. After intensive research, we found the following drawbacks to be consistent across multiple services:

- Limited number of free URLs
- Payment required for custom URLs
- Limited validity of URLs for free users
- Slow redirects
- Cannot view your old URLs
- Non interactive UI/UX for new users
- Cannot view stats about old URLs

Our service, TRIM aims to solve most of the above drawbacks. TRIM is a free service available for all. There is no barrier on the number of URLs that a user can shorten and the shortened URLs have a lifetime validity. The service can be availed as a Guest user as well and if a user wishes to view stats or his/her old URLs then the user can Sign In and shorten the URLs. The service makes use of Redis as a caching layer, in order to ensure faster redirects. TRIM has a very interactive UI/UX, to which a new user can adapt to very easily, without any difficulty. A logged in user can view stats like the number of clicks on each of their shortened URLs.

PROPOSED SYSTEM DESIGN

MODULE DESCRIPTION

1. **LOGIN PAGE:**
   It is a login interface where people can login as a registered user or can login as guest users. A guest user does not get all the facilities provided to the registered user. Some of these facilities may include viewing your stats or your shortened URLs.

2. **SIGNUP PAGE:**
   It is a signup interface where people can register with the portal by providing their valid email address and password. A registered user can then log in and avail to the facilities.

3. **TRIM PAGE:**
   This page is the main page where the basic functionality of our app takes place. This module has a text box where a user can put their link which they want to shorten and then when the link is shortened the user gets their shortened URLs along with the option to copy them. This module also provides the facility to generate custom URLs.

4. **ABOUT US PAGE:**
   This page consists of the basic information about the application. If a user wants to know more about the application, they can visit the about us page.

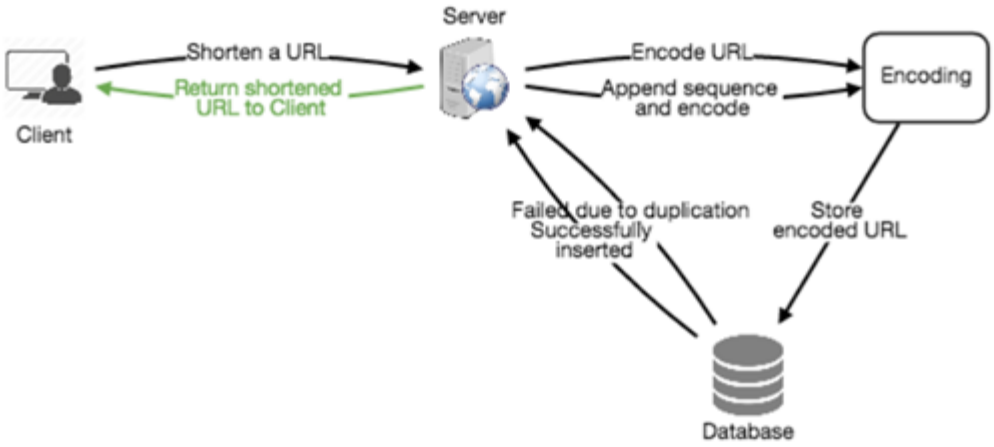
5. **FAQ PAGE:**
   This page is our FAQ page where general doubts of a user are cleared. A user can just go on this page and look for the basic solutions to their problem. The page has listed all the general questions that may come in the mind of the user.

6. **STATS PAGE:**
   This page is only visible to the registered users; people logged in as guest users will not be provided with this facility. This page shows the stats of no of links users have shortened. It shows how many links a user has shortened with the help of various graph

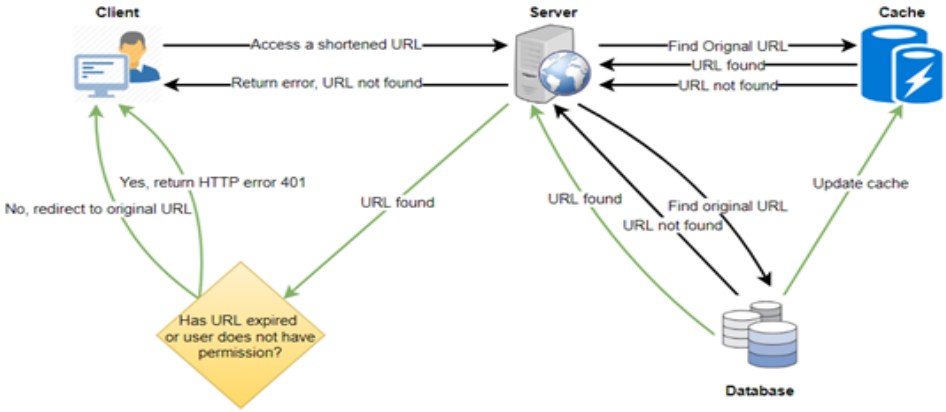
7. **LINKS PAGE:**
   This page is also only visible to the registered users; people logged in as guest users will not be provided with this facility. This page consists of a table with two columns one column has links which the user wanted to shorten while the other column has their respective shortened URLs. This page is very useful to the users as people can keep a track of all their shortened URLs.

DATA FLOW DIAGRAMS

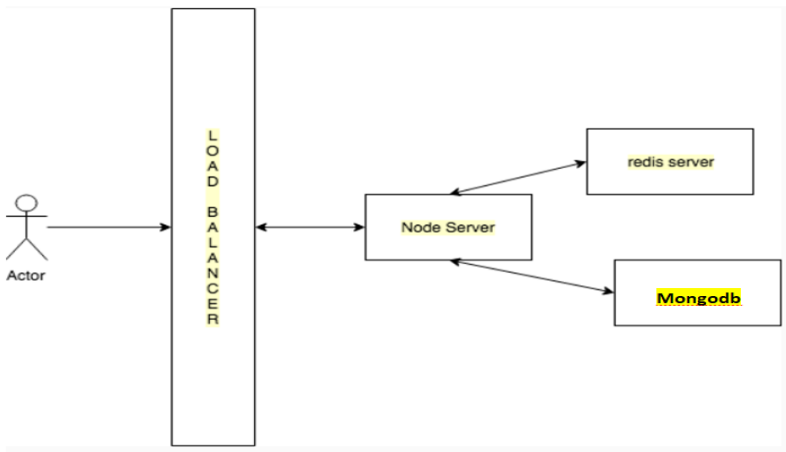
1. GENERATING SHORT URLS

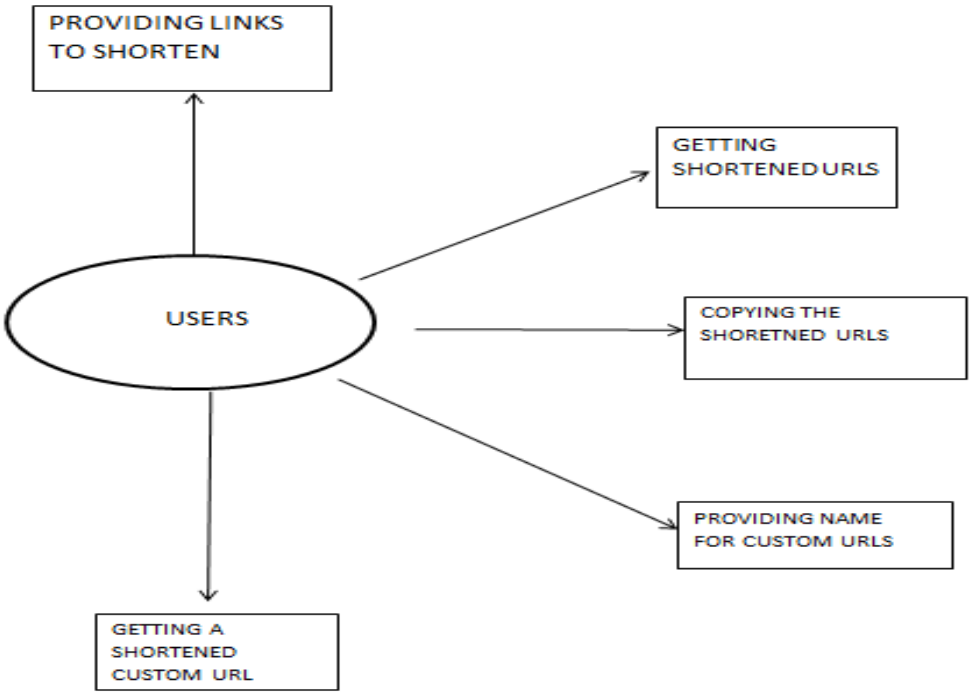


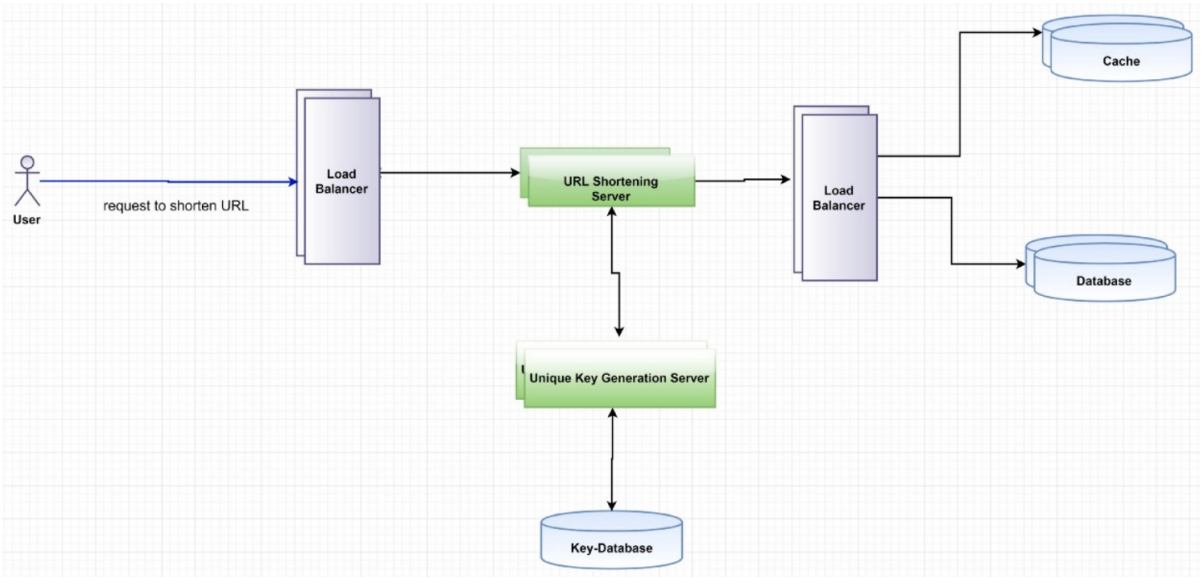
2. ACCESSING THE SHORTENED URLS:



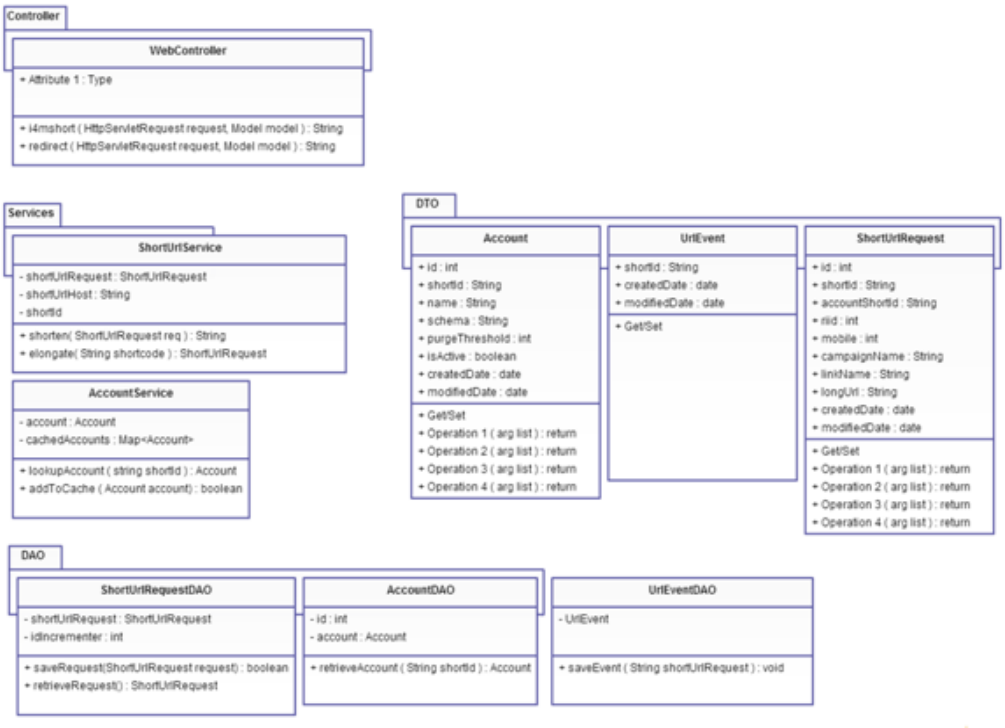
3. USE CASE DIAGRAMS



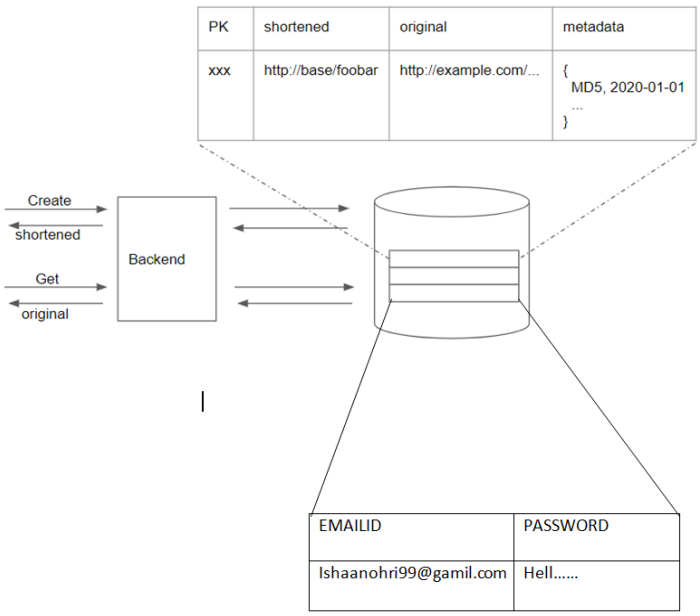

PROVIDING LINKS TO SHORTEN
GETTING SHORTENED URLS
USERS
COPYING THE SHORETNED URLS
PROVIDING NAME FOR CUSTOM URLS
GETTING A SHORTENED CUSTOM URL


User
request to shorten URL
Load Balancer
URL Shortening Server
Load Balancer
Cache
Database
Unique Key Generation Server
Key-Database

4. CLASS DIAGRAM



5. DATABASE DIAGRAM:



In the above diagram one can see the objects stored in the database such as username, password, shorthand, original URLs.

CODE SNIPPETS

shortener.ts

```
/* eslint-disable no-loop-func */
// eslint-disable-next-line no-unused-vars
import { Request, Response } from 'express';
import validator from 'validator';
import { code, message } from '../../config/messages';
import URL from '../../modals/URL';
import logger from '../../logger/logger-config';
import { getAsync, redisClient } from '../../app';

const cryptoRandomString = require('crypto-random-string');

const shortenURL = async (req: Request, res: Response) => {
  if (!req.body.custom && (!req.body.url || !req.body.email)) {
    res.status(400).send({
      success: false,
      code: code.wrongParameters,
      message: message.wrongParameters
    });
    return;
  }
  if (req.body.custom && (!req.body.url || !req.body.shortHand || !req.body.email)) {
    res.status(400).send({
      success: false,
      code: code.wrongParameters,
      message: message.wrongParameters
    });
    return;
  }

  let { url, shortHand, custom, email }: { url: string; shortHand: string; custom: boolean; email: string } = req.body;

  if (!validator.isURL(url)) {
    res.status(200).send({
      success: false,
      code: code.invalidURL,
      message: message.invalidURL
    });
    return;
  }

  if (url.includes(String(process.env.DOMAIN))) {
```

```
    res.status(200).send({
      success: false,
      code: code.alreadyShort,
      message: message.alreadyShort
    });
    return;
  }

  if (!custom) {
    let available: boolean = false;
    while (!available) {
      shortHand = cryptoRandomString({ length: 5, type: 'url-safe' });
      // eslint-disable-next-line no-await-in-loop
      if ((await getAsync(shortHand)) === null) {
        available = true;
      }
    }
  } else if (custom && !shortHand.match(/^([a-zA-Z0-9]+[-.~]?)+$/gm)) {
    res.status(200).send({
      success: false,
      code: code.invalidCustomURL,
      message: message.invalidCustomURL
    });
    return;
  }

  await redisClient.setnx(shortHand, url, async (err, reply) => {
    if (reply === 0) {
      res.status(200).send({
        success: false,
        code: code.shortHandUnavailable,
        message: message.shortHandUnavailable
      });
    } else {
      const urlDB = new URL({
        url,
        shortHand,
        email
      });
      try {
        await urlDB.save();

        res.status(201).send({
          success: true,
          url,
          shortHand: `${process.env.DOMAIN}/${shortHand}`
        });
      } catch (err2) {
        logger.error(err2);
        res.status(500).send({
          success: false,
```

```
                code: code.serverError,
                message: message.serverError
              });
            }
        }
    });
};

const redirect = async (req: Request, res: Response) => {
    const site = await getAsync(req.params.id);
    if (site !== null) {
        res.redirect(site);
    } else {
        res.redirect('/');
    }
};

export { shortenURL, redirect };
```

signup.html

```
<!DOCTYPE html>
<html>
    <head>
        <title>Sign-up Page</title>
        <style>
            .error {
                color: red;
                margin-left: 5px;
            }

            label.error {
                display: inline;
            }
        </style>

        <link
            rel="stylesheet"
            href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css"
            integrity="sha384-MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO"
            crossorigin="anonymous"
        />

        <link
            rel="stylesheet"
            href="https://use.fontawesome.com/releases/v5.3.1/css/all.css"
            integrity="sha384-mzrmE5qonljUremFsqc01SB46JvROS7bZs3IO2EmfFsd15uHvIt+Y8vEf7N7fWAU"
            crossorigin="anonymous"
        />
```

```
    <link rel="stylesheet" type="text/css" href="login.css" />

    <link href="https://maxcdn.bootstrapcdn.com/bootstrap/4.1.1/css/bootstrap.min.css" rel="stylesheet" id="bootstrap-
css" />
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.1.1/js/bootstrap.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
  </head>

  <body>
    <div class="container">
      <div class="d-flex justify-content-center h-100">
        <div class="card">
          <div class="card-header">
            <h3>Sign Up</h3>
            <div class="d-flex justify-content-end social_icon">
              <span><i class="fab fa-facebook-square"></i></span>
              <span><i class="fab fa-google-plus-square"></i></span>
              <span><i class="fab fa-twitter-square"></i></span>
            </div>
          </div>
          <div class="card-body">
            <form name="myForm" action="https://iwp-project.ml/signUp" method="post">
              <div class="input-group form-group">
                <div class="input-group-prepend">
                  <span class="input-group-text"><i class="fas fa-envelope"></i></span>
                </div>
                <input type="email" class="form-control" placeholder="email" id="email" required />
              </div>
              <div class="input-group form-group">
                <div class="input-group-prepend">
                  <span class="input-group-text"><i class="fas fa-key"></i></span>
                </div>
                <input type="password" class="form-control" placeholder="password" id="password" required />
              </div>
              <div class="input-group form-group">
                <div class="input-group-prepend">
                  <span class="input-group-text"><i class="fas fa-key"></i></span>
                </div>
                <input type="password" class="form-control" placeholder="Repeat password" id="repassword" required
/>

              </div>
              <div class="row align-items-center remember"><input type="checkbox" />Remember Me</div>
              <div class="form-group">
                <input type="submit" value="Sign-up" class="btn float-right login_btn" id="btnnLogin" />
              </div>
            </form>
          </div>
        </div>
      </div>
    </div>
```

```
    <script src="signup.js"></script>
  </body>
</html>
```

signup.js

```
$(document).ready(function () {
  $('#btnnLogin').click(function (e) {
    e.preventDefault();
    var email = $('#email').val();
    var password = $('#password').val();
    var settings = {
      url: 'https://iwp-project.ml/signUp',
      method: 'POST',
      timeout: 0,
      headers: {
        'Content-Type': 'application/json'
      },
      data: JSON.stringify({ email: email, password: password })
    };
    $('.error').remove();
    if (email.length < 1) {
      $('#email').after('<span class="error">This field is required</span>');
    } else {
      var regEx = /^[a-z0-9._-]+@[gmail]+.[com]{2,5}$/i;
      var validEmail = regEx.test(email);
      if (!validEmail) {
        $('#email').after('<span class="error">Enter a valid email</span>');
      }
    }
    if (password.length < 8) {
      $('#password').after('<span class="error">Password must be at least 8 characters long</span>');
    } else {
      $.ajax(settings).done(function (response) {
        console.log(response);
        window.open('index.html', '_self');
      });
    }
  });
});
```

login.css

```
@import url('https://fonts.googleapis.com/css?family=Numans');

html,
body {
```

```
  background-image: url('https://www.itl.cat/pngfile/big/73-737396_android-material-wallpaper-graphics.png');

  height: 100%;
  font-family: 'Numans', sans-serif;
}
.error {
  color: red;
  margin-left: 5px;
}

label.error {
  display: inline;
}
.container {
  height: 100%;
  align-content: center;
}

.card {
  height: 370px;
  margin-top: auto;
  margin-bottom: auto;
  width: 400px;
  background-color: rgba(0, 0, 0, 0.5) !important;
}

.social_icon span {
  font-size: 60px;
  margin-left: 10px;
  color: rgb(16, 229, 152);
}

.social_icon span:hover {
  color: white;
  cursor: pointer;
}

.card-header h3 {
  color: white;
}

.social_icon {
  position: absolute;
  right: 20px;
  top: -45px;
}

.input-group-prepend span {
  width: 50px;
  background-color: rgb(16, 229, 152);
  color: black;
```

```
    border: 0 !important;
}

input:focus {
    outline: 0 0 0 0 !important;
    box-shadow: 0 0 0 0 !important;
}

.remember {
    color: white;
}

.remember input {
    width: 20px;
    height: 20px;
    margin-left: 15px;
    margin-right: 5px;
}

.login_btn {
    color: black;
    background-color: rgb(16, 229, 152);
    width: 100px;
}

.login_btn:hover {
    color: black;
    background-color: white;
}

.links {
    color: white;
}

.links a {
    margin-left: 4px;
}
```

RESULTS

The website is completely up and running at https://iwp-project.ml/index.html and can be accessed from any device.

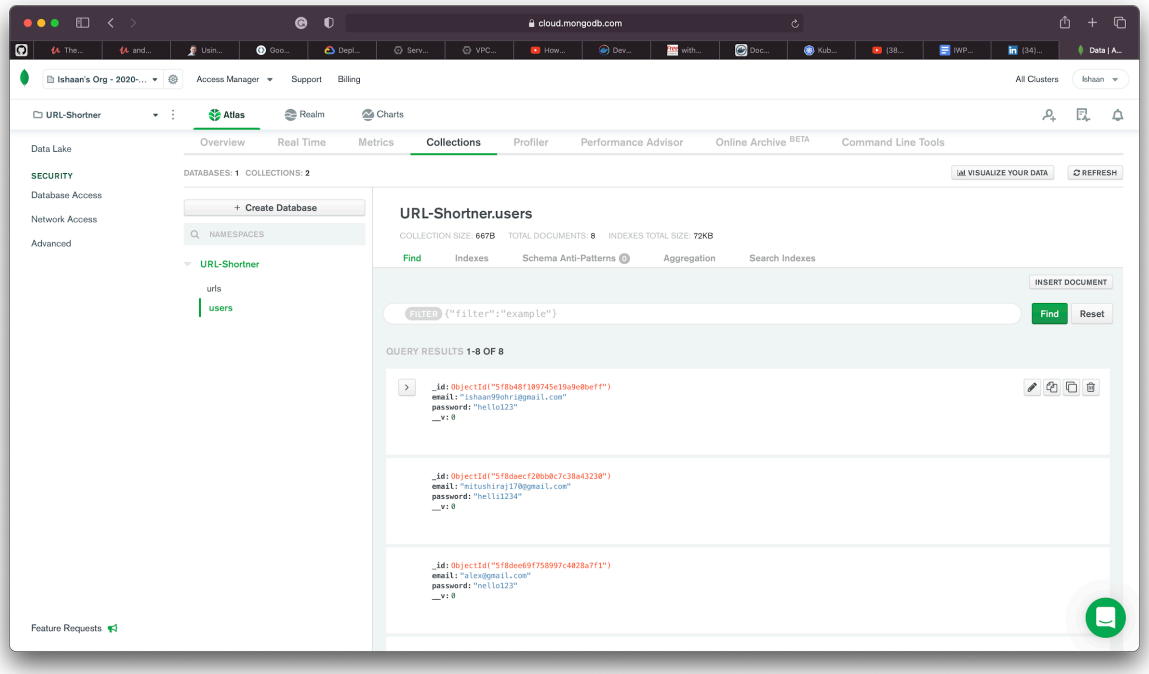
The code involving both frontend and backend modules can be viewed at https://github.com/IshaanOhri/IWP-Project

The service makes use of MongoDB as the database. The database is accessible via MongoDB Atlas and is hosted over AWS. MongoDB is a NoSQL database which makes use of collections instead of tables and documents instead of rows.

The database has two collections, one for user details and the other for URL details. Below are the Schema for the two collections along with respective screenshots.

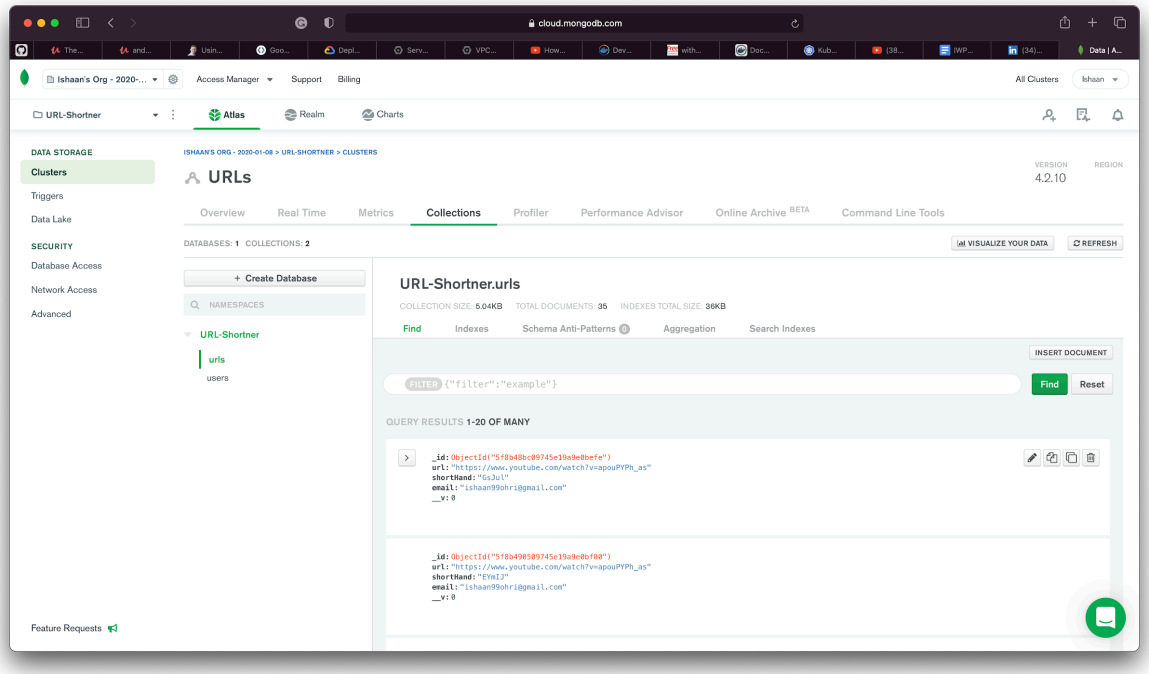
USER DETAILS:

Field Name	Data Type
_id (Primary Key)	MongoDB ObjectID
email	String
password	String



URL DETAILS:

Field Name	Data Type
_id (Primary Key)	MongoDB ObjectID
url	String
shortHand	String
email	String

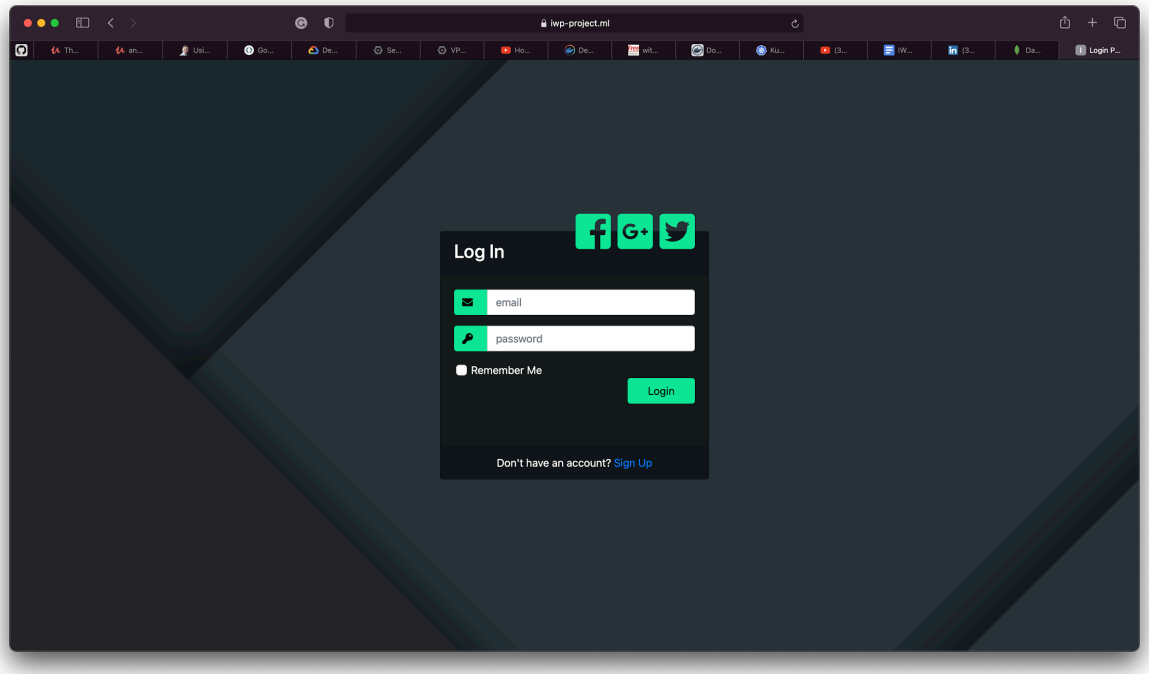


MongoDB Atlas takes care of automatic sharding, incorporates master slave concept for read and write of data. It also takes care of regular database backups, database maintenance and auto scaling.

FRONTEND MODULES:

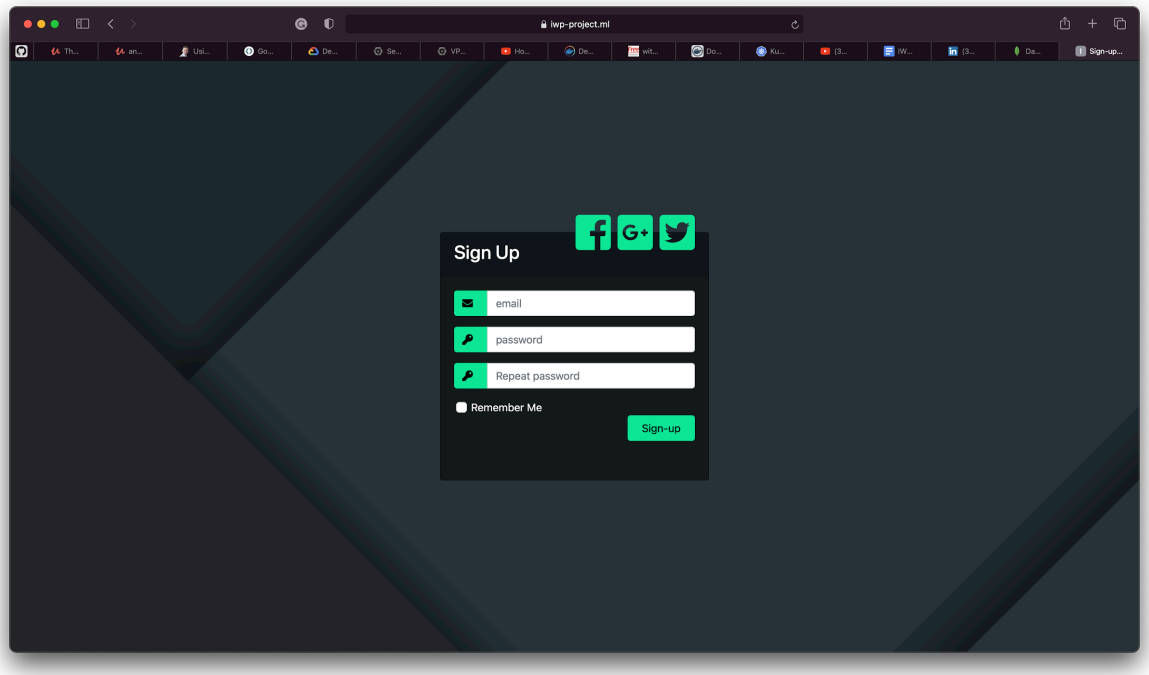
Module 1: Login page

Here people can log in as registered users or as guest users:

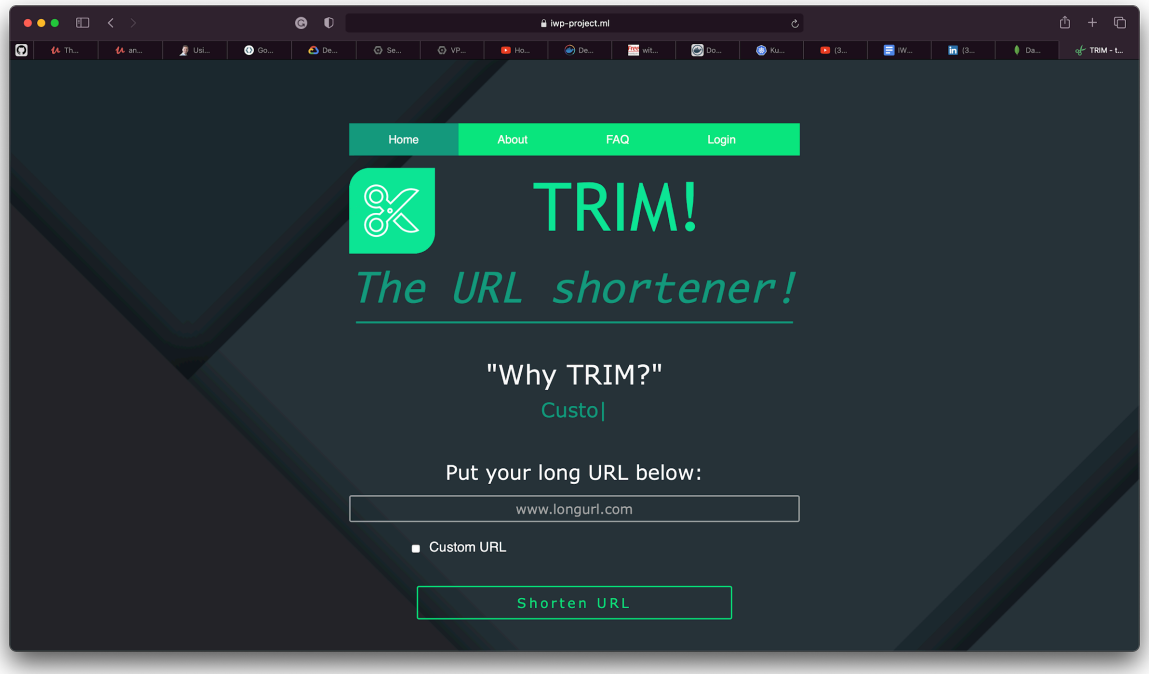


Module 2: Sign up Page

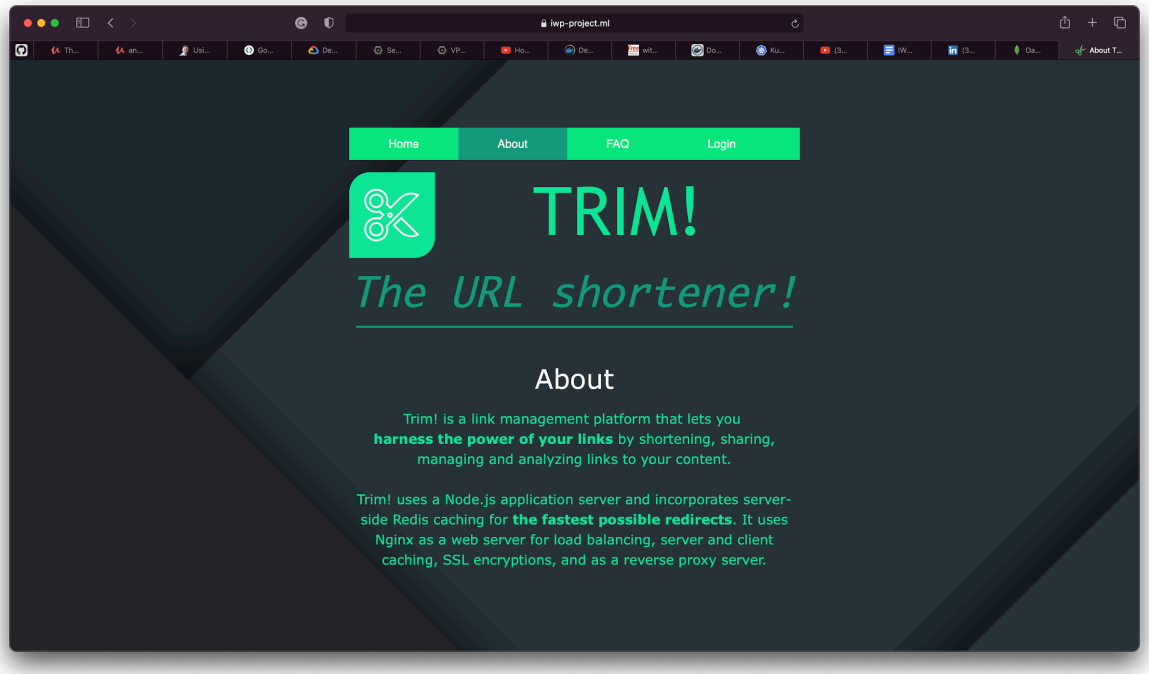
On this page people can register to the app using their Gmail accounts and password.



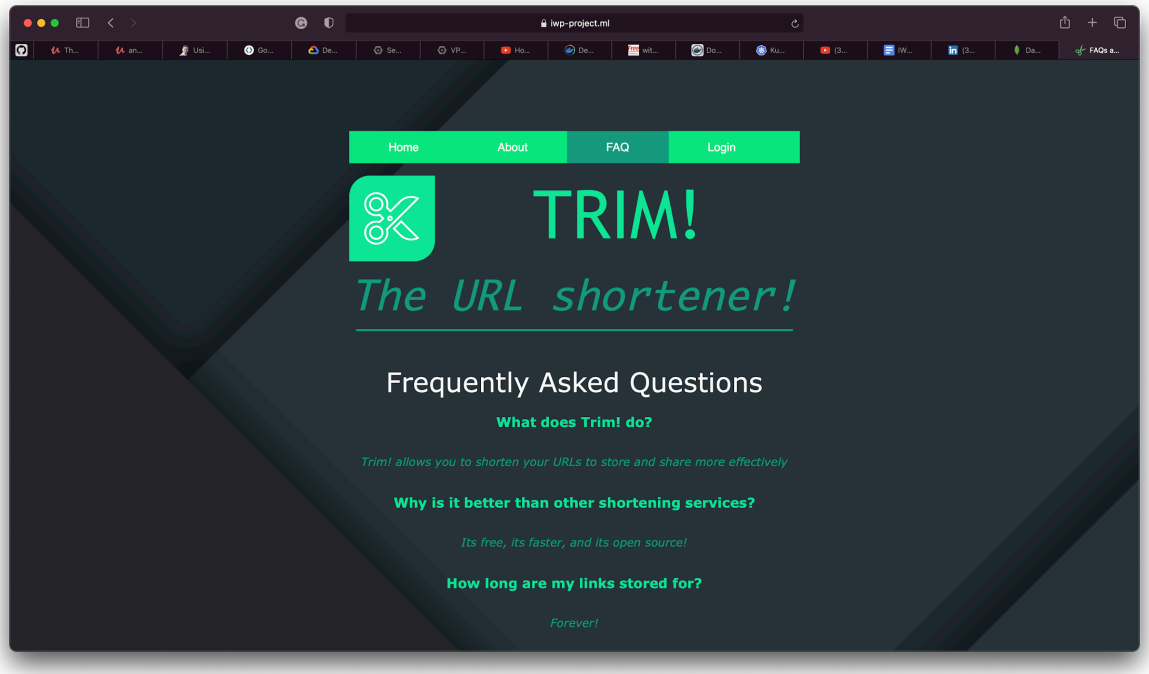
Module 3: Trim Page:
People can shorten their URLs here



Module 4: About Us Page:
Users can find the information related to this application on this page.

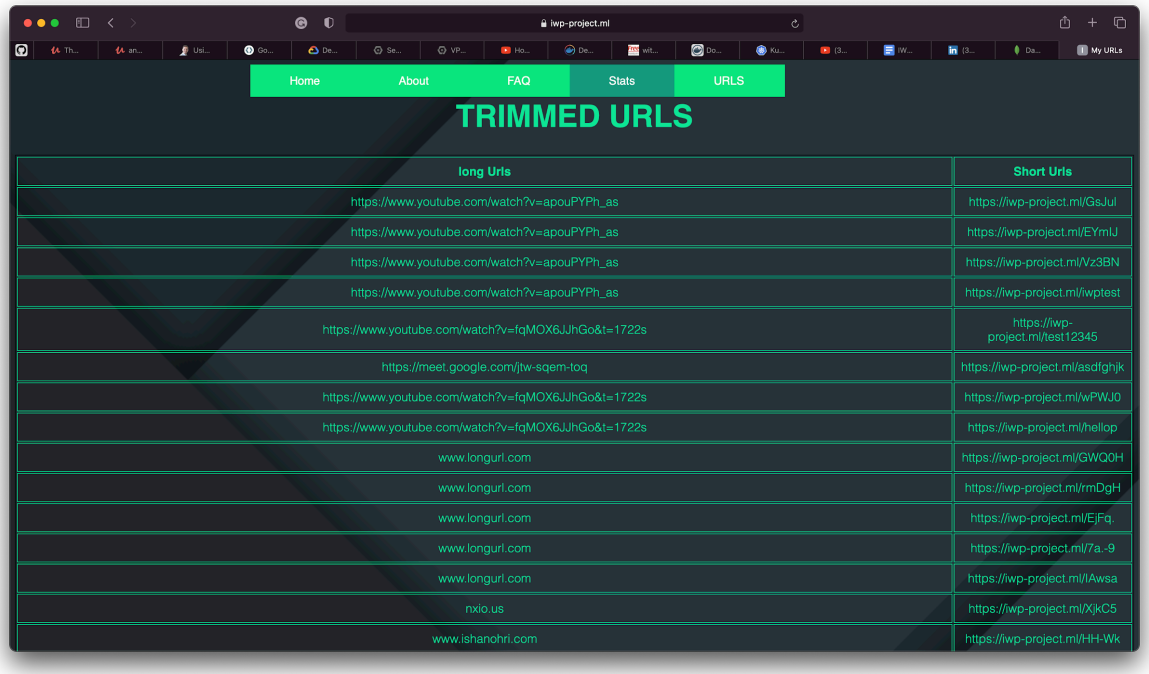


Module 5: FAQ Page:
Frequently asked questions are put up here



Module 6: Stats Page
Users can see their activities via graph

Module 7: Links Page
Users can find all their shortened URLs here



CONCLUSION

We've successfully completed our problem statement, made a fully functioning URL trimming website that's:

- Faster
- Simpler
- More easily accessible
- More aesthetic

than pre-existing services by making full use of multiple languages, tools and frameworks learnt during the course of this project.

REFERENCES

[1] Demetris Antoniades, Iasonas Polakis and Georgios Kontaxis;" we.b: The web of short URLs", Hyderabad,India March 28 – April 1, 2011.

[2] F. Klien, M. Strohmaier. Short Links Under Attack: Geographical Analysis of Spam in a URL Shortener Network. In proceedings of the 23rd ACM conference on Hypertext and social media (2012), Pages 83-88.

[3] D. Antoniades, I. Polakis, G. Kontaxis. we.b: The web of short URLs. In proceedings of the 20th international conference on the World wide web (2011), Pages 715-724.

[4] Wikipedia, URL redirection, http://en.wikipedia.org/wiki/URL_redirection

[5] V. Kandylas and A. Dasdan. The Utility of Tweeted URLs for Web Search. In proceedings of the 19th international conference on World wide web (2010), Pages 1127-1128.

[6] K. Thomas, C. Grier, J. Ma, V. Paxson, and D. Song. Design and Evaluation of a Real-Time URL Spam Filtering Service. Security and Privacy (SP) IEEE Symposium (2011), Pages 447 - 462.

[7] Brin, S., Page, L.: The anatomy of a large-scale hypertextual web search engine. Comput. Networks and ISDN Systems 30, 107–117 (1998)

[8] F. Benevenuto, G. Magno, T. Rodrigues, and V. Almeida. Detecting Spammers on Twitter. In Collaboration, Electronic messaging, AntiAbuse and Spam Conference (CEAS) (2010).

[9] Alexander Neumann ; "Analyzing Security Implications of URL Shortening Services", Diploma Thesis, RWTH Aachen University - Research Group IT-Security, 2011.

[10] A. Neumann, J. Barnickel, U. Meyer. Security and Privacy Implications of URL Shortening Services. In proceedings of Web 2.0 Security and Privacy (W2SP) (2011).