

***A Mini Project Report
on***

DevBootCamp

T.E. - I.T Engineering

Submitted By

Anuj Kundar 20104047

Chirag Padyal 20104034

Vishal Bangar 20104084

Under The Guidance Of

Mrs. Rujata Chaudhari



DEPARTMENT OF INFORMATION TECHNOLOGY

A.P. SHAH INSTITUTE OF TECHNOLOGY

G.B. Road, Kasarvadavali, Thane (W), Mumbai-400615

UNIVERSITY OF MUMBAI

Academic year: 2022-23

CERTIFICATE

This to certify that the Mini Project report on DevBootCamp has been submitted by Chirag Padyal (20104034), Vishal Bangar (20104084) and Anuj Kundar (20104047) who are a Bonafede students of A. P. Shah Institute of Technology, Thane, Mumbai, as a partial fulfilment of the requirement for the degree in **Information Technology**, during the academic year **2022-23** in the satisfactory manner as per the curriculum laid down by University of Mumbai.

Mrs. Rujata Chaudhari
Guide

Dr. Kiran Deshpande
Head Department of Information Technology

Dr. Uttam D.Kolekar
Principal

External Examiner(s)

- 1.
- 2.

Place: A.P.Shah Institute of Technology, Thane

Date:

ACKNOWLEDGEMENT

This project would not have come to fruition without the invaluable help of our guide Mrs. Rujata Chaudhari. Expressing gratitude towards our HoD, **Dr. Kiran Deshpande**, and the Department of Information Technology for providing us with the opportunity as well as the support required to pursue this project. We would also like to thank our teacher Mrs. Rujata Chaudhari who gave us her valuable suggestions and ideas when we were in need of them. We would also like to thank our peers for their helpful suggestions.

ABSTRACT
TABLE OF CONTENTS

1. Introduction.....	01
1.1. Purpose.....	01
1.2. Problem Statement.....	01
1.3. Objectives.....	01
1.4. Scope.....	02
2. Literature Review.....	03
3. Proposed System.....	04
3.1. Features and Functionality.....	04
4. Requirements Analysis.....	05
5. Project Design.....	06
5.1. Block diagram.....	06
5.2. DFD.....	07
6. Technical specification.....	10
7. Project Scheduling.....	11
8. Implementation.....	13
9. Result and Discussion.....	16
10. Conclusion and Future Scope.....	19

References

Chapter 1

Introduction

After this covid period E-Learning has become an important part of our education system, So we introduce DevBootCamp which is an online course provider for Aspiring Software developers.

DevBootCamp offers online courses, certifications, and degrees in various Tech Fields.

The Problem we identify is, Users need a website to track their work and create a routine that can maximize their workout.

The Solution we proposed is, The user will be able to track work for a particular day with the ability to add exercises. Each exercise can have a corresponding detail of sets and information about how to do the corresponding exercise.

1.1 Purpose

Providing life-transforming learning experiences to learners around the world.

1.2 Problem Statement

Nowadays learners don't have a specific book for learning new emerging and complex technologies to keep themselves updated with new tech stack and get better opportunities This web application will help users by providing learning resources

1.3 Objective

- To empower people to learn new and complex frameworks.

We achieve it by providing new and relevant courses and also allow new enthusiastic instructor to create relevant courses.

- To make best use of user's time

We provide game like learning experience so that learner can make use of their time pro-efficiently.

- To add new skills in user's arsenal

We achieve it by providing new and relevant courses and also allow new enthusiastic instructor to create relevant courses.

- To build a user-friendly website.

Simple and efficient user-friendly for beginner user.

- To track users learning progress.

We provide a efficient learning tracker.

- To track their workout progress with the help of graph.

We provide graphs to see the user growth over time

- To be up to date regarding new Techstacks and new technologies through articles.

We provide graphs to see the user growth over time

Scope

- Can be used to track course progression.
- Can be used to add exercises to workouts and their corresponding sets.
- Can be used to understand technical concepts through videos.
- Can be used to see improvement through quiz after completion of sections.
- Can be used to have a news of various articles regarding new technologies.

Chapter 2

Literature Review

Gaurav Chopra, Pankaj Madan and Preeti Bhanskar published the “OnlineTeaching and Learning in Higher Education during the Coronavirus Pandemic” in “Interactive Technology and Smart Education” conference on 2020 and according to them the major advantages of their system is Flexibility, Time Management, Skill Development, Wider Range of Courses and Programs, they used following techstack nginx, mysql , nodejs, django, aws ec 2 , react, bootstrap and graphql. They also came across various limitation of their systsm that is Sample was non-probabilistic and the research was conducted only on two Romanian universities. Thus, the results can not be generalized to the entire Romanian higher education. From this publication we concluded to use their robust tech stack as it provide scalability that is react, nginx and flask.

Chengfeng Xue and Xiting Zeng published the “Design of a Web-based Personalized E-learning Platform” in “Journal of Physics Conference series” conference on 2014 and according to them the major advantages of their system is It is implemented in B/S mode, which is one of the key points of the system. Interactive ability of the system Network servicequality, they used following techstack nginx, mysql , nodejs, django, aws ec 2 , react, bootstrap and graphql. They also came across various limitation of their systsm that is system is e-learning create isolation. From this publication we concluded to use mysql database as it is cheap and profitable for production.

Amal Al-Dujaily published the “A Study on Personality in Designing Adaptive e-Learning Systems” in “Advanced Learning Technologies, 2008. ICALT '08. Eighth IEEE International” conference on 2008 and according to them the aim of this study is to understand how learners with different personality respond to an e-learning content structure, by which it will help to build an instance of personality traits in the design of an effective adaptive e-learning system. An experiment performed to explore the relationship between the learnerpsilas personality type and the learning sequence design. It revealed that a different personality type (i.e., extraverted vs. introverted) had a markedly different effect on learning performance. From this publication we concluded, how to improve relation of courses and its consumer i.e learner.

Chapter 3

Proposed System

The DevBootcamp is designed to assist people with insufficient knowledge about software development process. The system will feature learning resources and project building exercises. For the user the proposed system will benefit the users by giving technical knowledge about the development process.

3.1 Features and Functionality

- **Video Streaming of Courses:**

Users can stream videos of available courses for better understanding of concepts up to full HD resolutions. Users can watch courses according to their preference using playback speed option provided in video player.

- **Quiz /exercises:**

Users can test their grasp on knowledge by giving test/quiz after completing each section. Users can reattempt the quiz.

- **Detailed Solutions to problems:**

User can view the solution related to the exercises of that particular section to review their attempted quiz questions

- **Graphical Representation:**

User can observe his/her progress through the graph constantly which will show user how much progress is made in the enrolled course.

- **Track Progress:**

User can track their daily course progression. After completing each section it gets updated in progress bar. Completion Tracking allows instructors to set up a course site so that learners are easily able to see their progress. This feature allows for two main types of completion indicators: Activity Completion (for individual activities, such as assignments) and an overall Course Completion.

Course completion can be set up so that when students complete a series of requirements with the course they are marked as having completed the course. The criteria can include meeting an activity's grade level or a manual checking "complete" by either the student and/or teacher. The report can also show if the student has completed another course(s) that is marked as a "completion dependent" course.

Chapter 4

Requirement Analysis

4.1. Performance Requirements

The load time for the user interface screen should take no longer than 5 seconds.

Course videos for reference should be there.

Track of daily progress made by user should be seen through graph or daily tracking information.

4.2. Design Constraints

The application should be able to run on any Pc or Laptop.

4.3. Availability

The application should be available at all times whenever user wants to use.

4.4. Hardware requirements

1. RAM

The application requires a device with a minimum of 512MB RAM while running.

2. Processor speed

The application requires a device with a minimum processor speed of 1GHz while running.

4.5. Software requirements

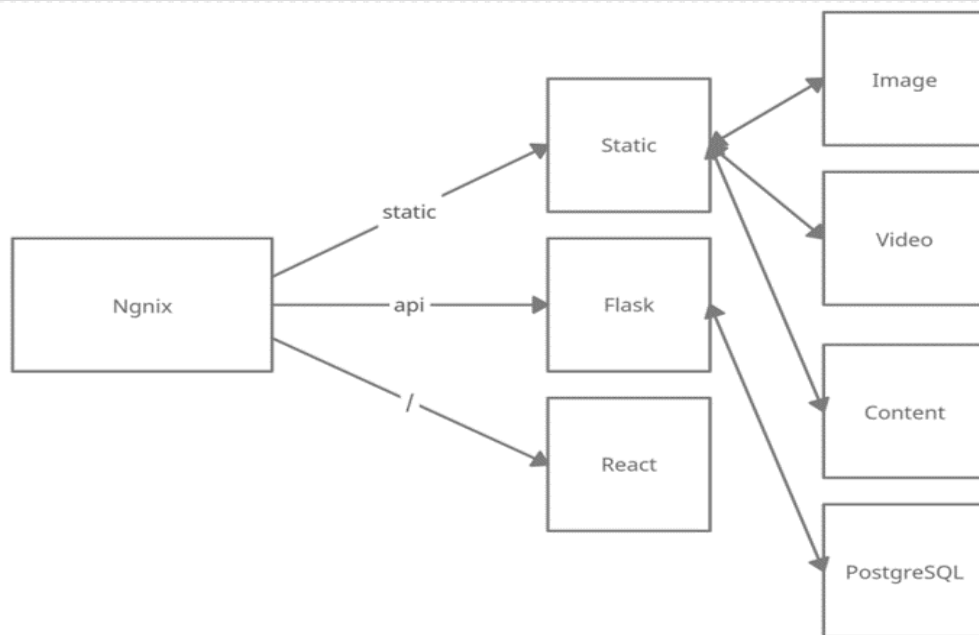
Operating system

The application must run on any Operation System.

Chapter 5

Project Design

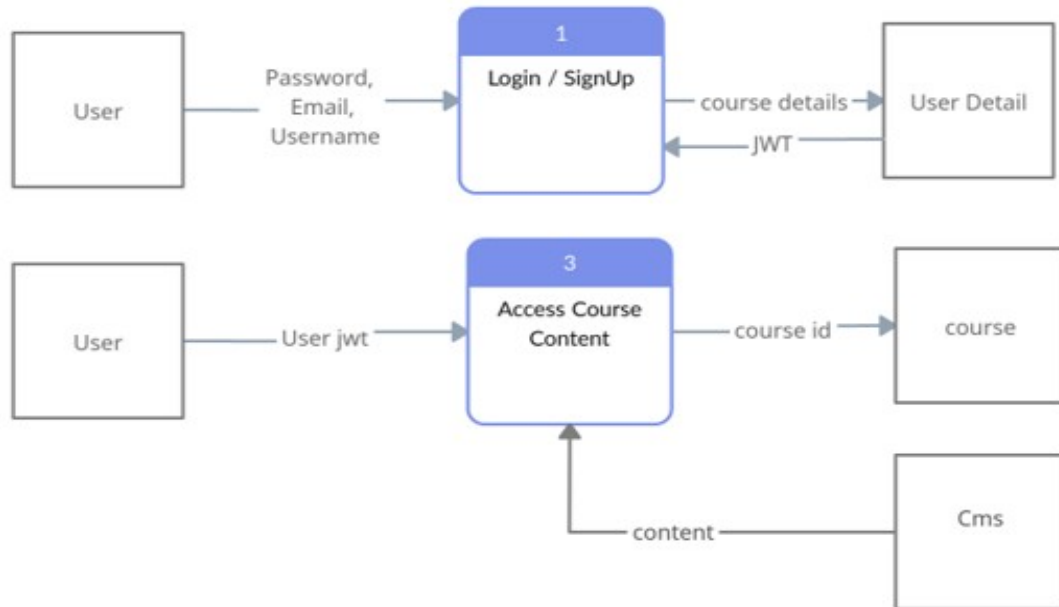
5.1 Block Diagram



5.1) DFD for Login and Course Access

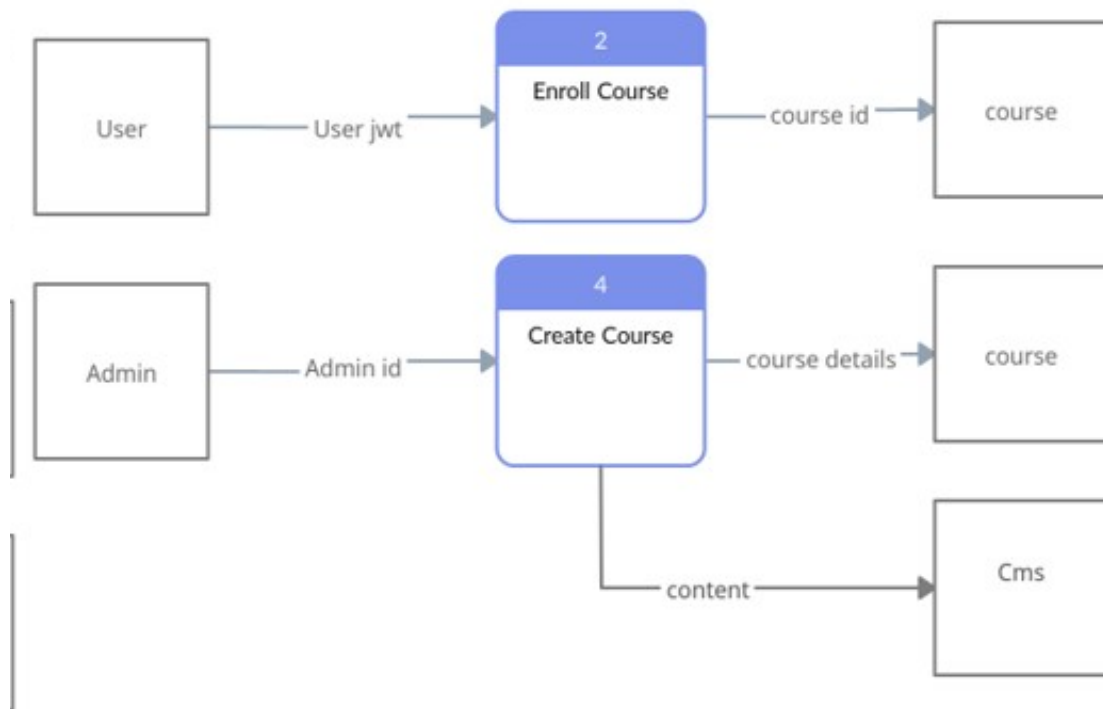
We are using nginx as our webserver, so its a base of our system architecture. Nginx maps the routes to specific infrastructure to have a proper data flow. Static files like images, videos or other multimedia files are served using “/static” route. And any backend api query or request is handled using “/api” routes it avoid the cors errors and we can host front-end and backend on a same device or server. All the remaing routes excluding \static and \api are served under react. Flask also has postgresql which is our database for normal data entries, any larger binaries get stored in server and course content is served through \static routes by provided cms (content management system)

5.2 DFD



5.2.1) DFD for Login and Course Access

In the above dfd diagram the user must login or register in our server then a jwt or json web token is generated. With the help of jwt the user can access the course detail page. The Jwt is also required to access course content or register a course. The Jwt lasts for 30 minutes, so each user session last for 30 minutes maximum, increasing security.



5.2.2) DFD for Enroll course and Create course

In the above dfd diagram the user must login or register in our server then a jwt or json web token is generated. With the help of jwt the user can access the course detail page.

The Jwt is also required to access course content or register a course.

The admin doesn't use jwt but has basic auth only, the admin can create course, once course created the course textual data will be stored in course database and course content to cms (content management system)

Chapter 6

Technical Specification

6.1. Front-end: -

1. Framework: - React
2. State Manager: - Redux

6.2. Back-end: -

1. Framework: - Flask
2. Webserver: - Ngnix
3. Auth: - Basic Auth
4. Security: - Json Web Token

6.3. Database: - PostgreSQL.

Chapter 7

Project Scheduling

Sr. No	Group Member	Time duration	Work to be done
<u>1</u>	Anuj kundar Vishal Bangar Chirag Padyal	1 st week of August	Implementing Login, Registration and Home Page in react js.
		2 nd week of August	Implementing Flask Backend for Registration using Basic Auth and jwt and integrating it with front end
<u>2</u>	Anuj kundar Vishal Bangar Chirag Padyal	3 rd week of August	Implementing Rest full api in flask to handle different routes to perform course crud operations
		4 th week of August	Course Categories page frontend and itegrating with api
<u>3</u>	Anuj kundar Vishal Bangar Chirag Padyal	1 st week of September	Course Information page frontend and itegrating with api
		2 nd week of September	Course Content page and itegrating with api
<u>4.</u>	Anuj kundar Vishal Bangar Chirag Padyal	3 rd week of September	Adding course quiz funtion to course content
		3 rd week of September	Adding automatic certification generation
<u>5.</u>	Anuj kundar Vishal Bangar Chirag Padyal	4 th week of September	Serving Frontent and Backend on Nginx Web Server
		4 th week of September	Dockerizing the whole project

GANTT CHART TEMPLATE

Smartsheet Tip →

A Gantt chart's visual timeline allows you to see details about each task as well as project dependencies.

PROJECT TITLE DevBootCamp - Coursera clone
PROJECT GUIDE Prof. Rutika Chaudhari

INSTITUTE & DEPARTMENT ABIT (IT Dept.)
DATE 10/14/22

WBS NUMBER	TASK TITLE	TASK OWNER	START DATE	DUE DATE	DURATION WEEKS	PERCENTAGE OF TASK COMPLETE	PHASE ONE																								
							WEEK 1					WEEK 2					WEEK 3					WEEK 4					WEEK 5				
							M	T	W	R	F	M	T	W	R	F	M	T	W	R	F	M	T	W	R	F	M	T	W	R	F
1	Project Conception and Initiation																														
WEEK 1	Group formation and Topic finalization. Identifying the scope and objectives of the Mini Project	Chirag, Anvi, Vishal	7/20/22	2/3/22	1	100%																									
WEEK 2	Identifying the technology stack and functionalities of the Mini Project	Chirag, Anvi, Vishal	7/24/22	2/13/22	1	100%																									
WEEK 3	Discussing the project topic with the help of paper prototype and verified by guide	Chirag, Anvi, Vishal	7/27/22	2/22/22	1	100%																									
WEEK 4	Project Review I	Chirag, Anvi, Vishal	*9/14/22	3/16/22	1	100%																									
2	Project Design and Implementation																														
WEEK 5	Designing the Graphical User Interface	Anvi, Vishal	10/15/22	3/4/22	2	100%																									
WEEK 6	PostgreSQL Database Design and Core API Integration, Course crud operation, Validation and Authentication	Chirag, Anvi, Vishal	*10/24/22	3/18/22	2	100%																									
WEEK 7	Results and Report Writing	Anvi, Vishal	10/1/22	3/28/22	1	100%																									
WEEK 8		Anvi, Vishal	10/7/22	4/4/22	1	100%																									
WEEK 9	Project Review II	Chirag, Anvi, Vishal	10/14/22	4/12/22	1	100%																									

7.1) Timeline chart

8. Implementation

8.1) JWT implementation

```
def token_required(f):
    @wraps(f)
    def decorator(*args, **kwargs):
        token = None
        if 'x-access-tokens' in request.headers:
            token = request.headers['x-access-tokens']

        if not token:
            # return jsonify({'message': 'a valid token is missing'})
            return make_response(jsonify({'message': 'a valid token is missing'}), 404)

        try:
            data = jwt.decode(
                token, app.config['SECRET_KEY'], algorithms=["HS256"])
            current_user = Users.query.filter_by(
                public_id=data['public_id']).first()
        except:
            # return jsonify({'message': 'token is invalid'})
            return make_response(jsonify({'message': 'token is invalid'}), 404)

        kwargs['current_user'] = current_user
        return f(current_user, *args, **kwargs)
    return decorator
```

8.2) Email Verification

```
def generate_confirmation_token(email):
    serializer = URLSafeTimedSerializer(app.config['SECRET_KEY'])
    return serializer.dumps(email, salt=app.config['SECURITY_PASSWORD_SALT'])

def confirm_token(token, expiration=3600):
    serializer = URLSafeTimedSerializer(app.config['SECRET_KEY'])
    try:
        email = serializer.loads(
            token,
            salt=app.config['SECURITY_PASSWORD_SALT'],
            max_age=expiration
        )
    except:
        return False
    return email
```

8.3) Payment webhooks

```
@app.route('/api/pay', methods=['POST'])
@token_required
@check_confirmed
def pay(current_user):
    data = request.get_json()
    email = data['email']
    amount = data['amount']
    print("amount : %s" % (amount))

    if not email:
        return 'You need to send an Email!', 400

    intent = stripe.PaymentIntent.create(
        amount=int(amount),
        currency='inr',
        receipt_email=email
    )

    return {"client_secret": intent["client_secret"]}, 200


@app.route('/webhook', methods=['POST'])
def webhook():
    payload = request.get_data()
    sig_header = request.headers.get('Stripe-Signature', None)

    if not sig_header:
        return 'No Signature Header!', 400

    try:
        event = stripe.Webhook.construct_event(
            payload, sig_header, endpoint_secret
        )
    except ValueError as e:
        # Invalid payload
        return 'Invalid payload', 400
    except stripe.error.SignatureVerificationError as e:
        # Invalid signature
        return 'Invalid signature', 400

    if event['type'] == 'payment_intent.succeeded':
        # contains the email that will receive the receipt for the payment (users email usually)
        email = event['data']['object']['receipt_email']

        user_info['paid_50'] = True
        user_info['email'] = email
    else:
        return 'Unexpected event type', 400

    return "", 200
```

8.4) Nginx Configuration

```
upstream hello_flask {
    server web:5000;
}

server {

    listen 80;
    listen [::]:80;

    location / {
        root /usr/share/nginx/html;
        index index.html index.htm;
        try_files $uri $uri/ /index.html =404;
    }

    location ~ /api/ {
        proxy_pass http://hello_flask;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header Host $host;
        proxy_redirect off;
    }

    location /nginx/static/ {
        autoindex on;
        alias /home/app/web/static/;
    }

}
```

8.4) Environment Variables

```
export APP_SETTINGS="config.DevelopmentConfig"
export DATABASE_URL="postgresql://user:password@localhost:5432/devbootcamp"
export SECRET_KEY="004f2af45d3a4e161a7dd2d17fdae47f"
export STRIPE_PUBLISHABLE_KEY=""
export STRIPE_SECRET_KEY=""
export SECURITY_PASSWORD_SALT="my_precious_two"
export APP_MAIL_USERNAME=""
export APP_MAIL_PASSWORD=""
export APP_MAIL_ID=""
```

9. Result and Discussion

Frontend

Sign in to your account

Email address

anujkundar003@gmail.com

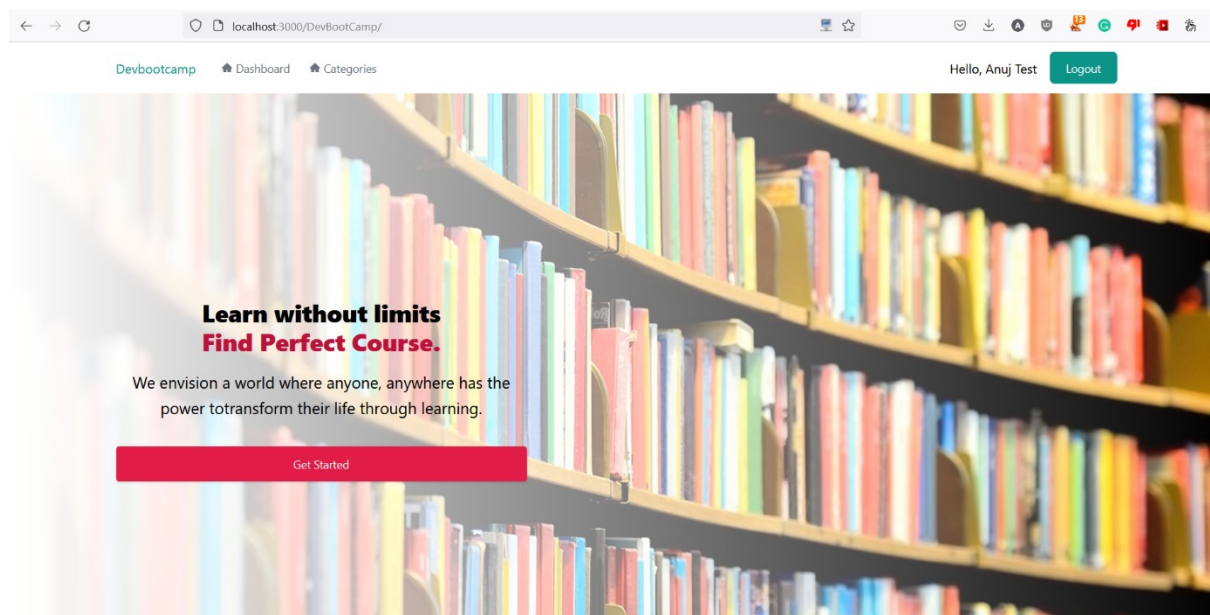
Password

••••••

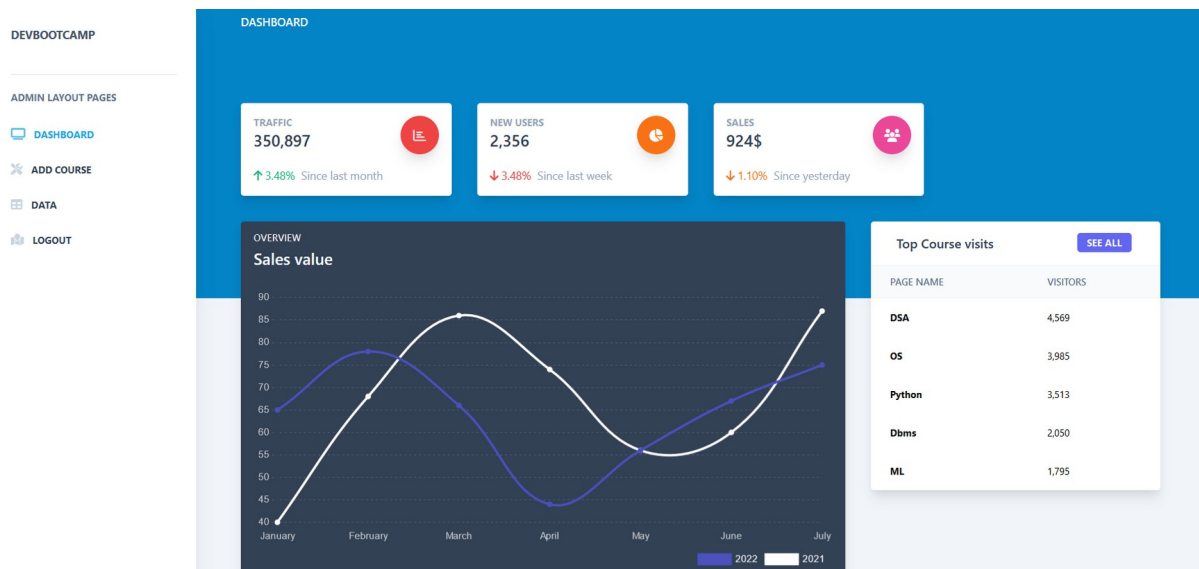
Sign in

Or Signup

9.1) Login Page



9.2) Home Page



9.3) Dashboard Page

DEVBOOTCAMP
ADMIN LAYOUT PAGES
DASHBOARD
ADD COURSE
DATA
LOGOUT

Add Course
ADD

COURSE INFORMATION

TITLE
Vue - The Complete Guide (incl. Router & Composition API)

CATEGORY
dev

TAG
dev

AUTHOR
Mr. Myagi

IMAGE
maxresdefault.jpg

DESCRIPTION
Vue.js is an awesome JavaScript Framework for building Frontend Applications! VueJS mixes the Best of Angular + React!

PRICE
30

LINE

RATING
1 2 3 4 5

AUTHOR IMAGE
Gon_Freecss.png
ADD

INSTRUCTOR PREVIEW

IMAGE PREVIEW

9.4) Add Course Page

Add Index

ADD SUBHEAD
ADD HEAD
SUBMIT

INDEX INFORMATION

HEADER

Basic

TITLE

Basic

MARKDOWN FILE

Browse... 1.md

MODULE QUIZ INFORMATION

JSON FILE

Browse... content.js

FINAL QUIZ FILE

JSON FILE

Browse... combineContent.js

9.5) Add Course Index Page

Add Index

ADD SUBHEAD
ADD HEAD
SUBMIT

INDEX INFORMATION

HEADER

TITLE

MARKDOWN FILE

Browse... 2.md

MODULE QUIZ INFORMATION

JSON FILE

Browse... content.js

FINAL QUIZ FILE

JSON FILE

Browse... combineContent.js

Index Tables

HEADER	TITLE
Intermediate	Basic
Intermediate	Intermediate
Intermediate	Quiz

9.6) Add Course Content Page

10. Conclusion and Future Scope

Online course portal project is developed in javascript (React) and Python (Flask). The main aim of this system is to implement an online based mostly portal with education data which is able to be helpful for faculty students. on-line education is one of the quick growing field on internet wherever users will directly solve issues by visiting web site with none facilitate from academics. This method had inflated scope of on-line education and on-line courses. in conjunction with on-line courses this computer are often updated with alternative data like job updates, on-line communication details, coming events, government job, and tips for teaching. The future scope of our system is :-

- Can be used in Education field to empower at-place learning.
- Can be used in Company's and industries to tech workers their latest tech, with no human error.
- Can be used by individual teacher, to spread its reach in global market.
- Can be used to see improvement through quiz after completion of sections.

Reference

- (1) Courses as markdown:- <https://github.com/remarkjs/react-markdown/>
- (2) Fetch Static (i.e. videos, markdownfiles, images):- <https://fedingo.com/how-to-use-nginx-with-flask/>
- (3) JWT :- <https://www.geeksforgeeks.org/using-jwt-for-user-authentication-in-flask/>
- (4) Dockerize:- <https://blog.miguelgrinberg.com/post/how-to-dockerize-a-react-flask-project>