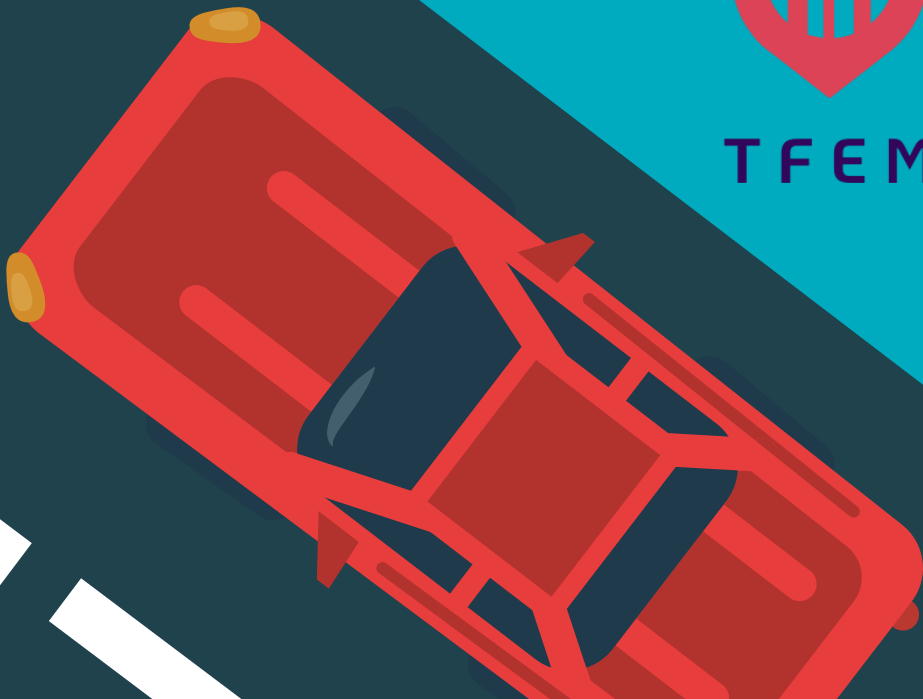




TFEM

Track Field Employee Movement

Fintech Essentials



Problem Statement

“Designing a Field Sales tracker to help optimize on cost and travel route. ”

We have to develop a product for Fintech which would help them to monitor their employee's location as well as suggest him/her the best cost effective route to their designated marketing locations.

Components

Requirements

- Designing a system by which company can track the movement of the field agent
- System should be mobile compatible, picks up the Geo coordinates of the device at periodic interval
- System should provide their route map with the distance covered everyday



For Hackathon

- We've assumed that the field sales person travels to 5 different locations in a day.
- The person spends x amount of time in each location with customer discussing business.
- The output required is distance travelled, and time spent in each of the locations.

Our Solutions



ATTENTION!

OPTIMUM DISTANCE

We suggest the employee the shortest route using the google APIs



TRACKING

We track the employees location and calculate the distance he has travelled the whole day



TIME

We calculate the time he spends at various locations



#3-Ds of Contribution



50%

Designing

Structural designing and
planning the whole
product



75%

Development

Working of the product
and its functionalities



25%

Debugging

Bug fixing and
modifications in the APIs
and keys

TFEM profile infographics

TECHNOLOGY

Using Python Django framework and Google APIs and HTML/CSS for frontend

COMPLETION

We have successfully made the frontend, were able to find the path and distance for the field staff



DESIGN

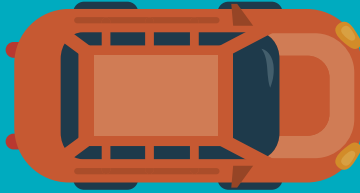
TFEM has been designed by taking into consideration the ease of usage

LEARNING

We never had experience with google APIs so we learnt it from scratch on how it is to be applied with Django

Working system infographics

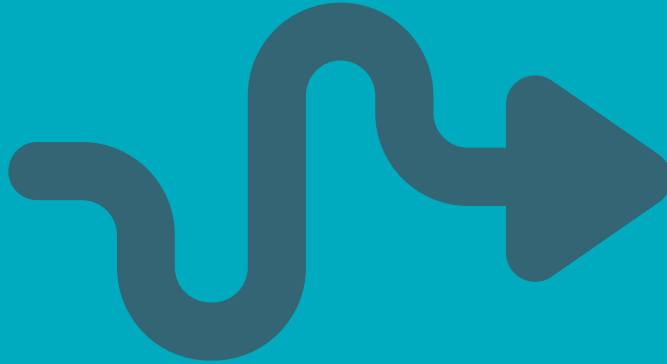
STEP 1



**VEHICLE TO BE
TRACKED**

We take the vehicle
number which needs
to be tracked

STEP 2



WAY

Find the location of
the vehicle

STEP 3



**IDENTITY OF
EMPLOYEE**

This happens by using
the identity of the
driver

Framework

01

METHODOLOGY

We have followed
Test-Driven
Development (TDD)
while making this
product

02

APIs

Places, Directions,
Distance matrix,
Geocoding, Maps
Javascript from
google APIs

03

FRONTEND

Used HTML, CSS,
bootstrap

04

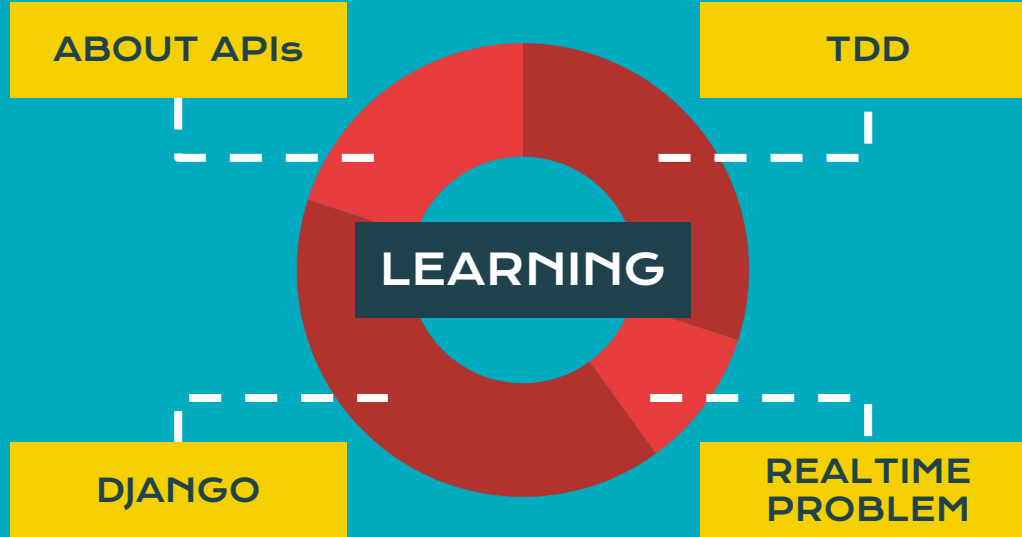
BACKEND

Django framework



We learnt how to use google APIs as it was a new thing for us to explore. It made our work easier specially in terms of usage of the tracking features.

It was a completely new framework for us to use which ensured all the features which any MERN stack website can do.



It made our whole development process easier and described the the expected behavior int the form of a test and then helped us in creating a code for the subproblems by refactoring.

We understood how to work under pressure in restricted time duration in a real time environment

INITIAL DESIGNS



TFEM

Sign In

Sign Up

P.V. Abhiram
20BAI1132
Dhruvi Ochani
20BCE1882
Nayan Khemka
20BCE1884
Shatakshi Shree
20BAI1314

E-cell Hackathon - 2022

Username

Password

☐ Show Passwords

Sign in

Field	
Start	52.3871067,0.1246447
Destination	52.3993973,0.2662897
Duration	25 minutes and 12 seconds
Distance	19.46 Km
Directions	click here

Map

Satellite

PS – We're working on the frontend and will come up with better implementation

FileEditSelectionViewGoRunTerminalHelp

settings.py - did_django_google_api_tutorial - Visual Studio Code

EXPLORER

did_django_google_api_tutorial

__pycache__

__init__.py

asgi.py

mixins.py

settings.py

urls.py

wsgi.py

env

main

__pycache__

migrations

__init__.py

admin.py

apps.py

models.py

tests.py

urls.py

views.py

static

branding

google_maps.js

google_places_waypoints.js

google_places.js

main.css

main.js

templates

users

db.sqlite3

LICENSE

OUTLINE

TIMELINE

settings.py

admin.py

google_places.js

did_django_google_api_tutorial > settings.py > ...

126

<https://docs.djangoproject.com/en/3.2/howto/static-files/>

127

128

STATICFILES_DIRS = [

129

os.path.join(BASE_DIR, 'static')

130

]

131

132

STATIC_URL = '/static/'

133

134

STATIC_ROOT = os.path.join(BASE_DIR, 'static_cdn')

135

136

GOOGLE_API_KEY = "AIzaSyDicmJ-M505THCYxPhNM4UDTZHjt395cxI"

137

138

RECAPTCHA_PUBLIC_KEY = "6LddMt0hAAAAAEYcmC2eJHq86ioM3D6qBot5BukR"

139

140

RECAPTCHA_PRIVATE_KEY = "6LddMt0hAAAAAFvKyPhVhmOPL0_lsd7mTptl9veE"

141

142

Default primary key field type

143

<https://docs.djangoproject.com/en/3.2/ref/settings/#default-auto-field>

144

145

DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'

146

147

LOGIN_URL = "users:sign-in"

148

LOGIN_REDIRECT_URL = "users:account"

149

LOGOUT_REDIRECT_URL = "users:sign-in"

150

151

BASE_COUNTRY = "UK"

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

JUPYTER

[07/Sep/2022 18:20:49]

"GET /static/main.js HTTP/1.1" 304 0

[07/Sep/2022 18:20:50]

"GET /route HTTP/1.1" 200 3573

[07/Sep/2022 18:20:50]

"GET /static/google_places_waypoints.js HTTP/1.1" 304 0

[07/Sep/2022 18:20:53]

"GET /map HTTP/1.1" 302 0

[07/Sep/2022 18:20:53]

"GET /route HTTP/1.1" 200 3573

[07/Sep/2022 18:20:55]

"GET / HTTP/1.1" 200 2683

[07/Sep/2022 18:20:56]

"GET /profile HTTP/1.1" 200 3674

[07/Sep/2022 18:20:56]

"GET /static/google_places.js HTTP/1.1" 304 0

[07/Sep/2022 18:20:58]

"GET / HTTP/1.1" 200 2683

main*

78°F Cloudy

Ln 136, Col 58

Spaces: 4

UTF-8

CRLF

Python

3.10.5 64-bit

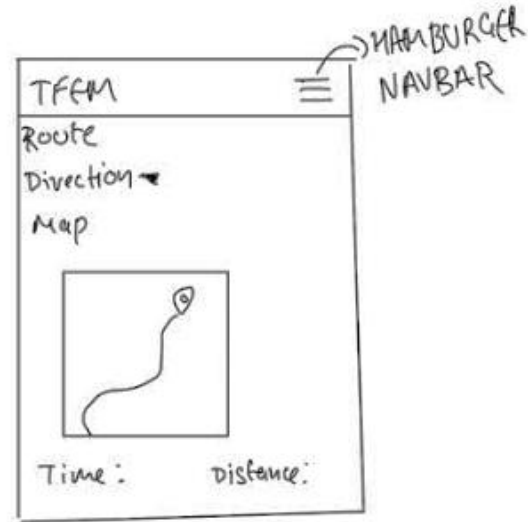
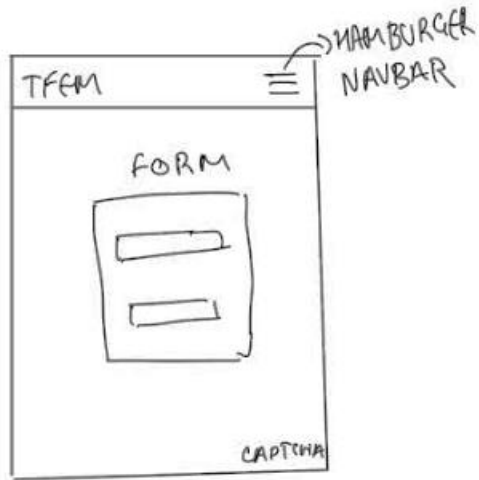
Go Live

Prettier

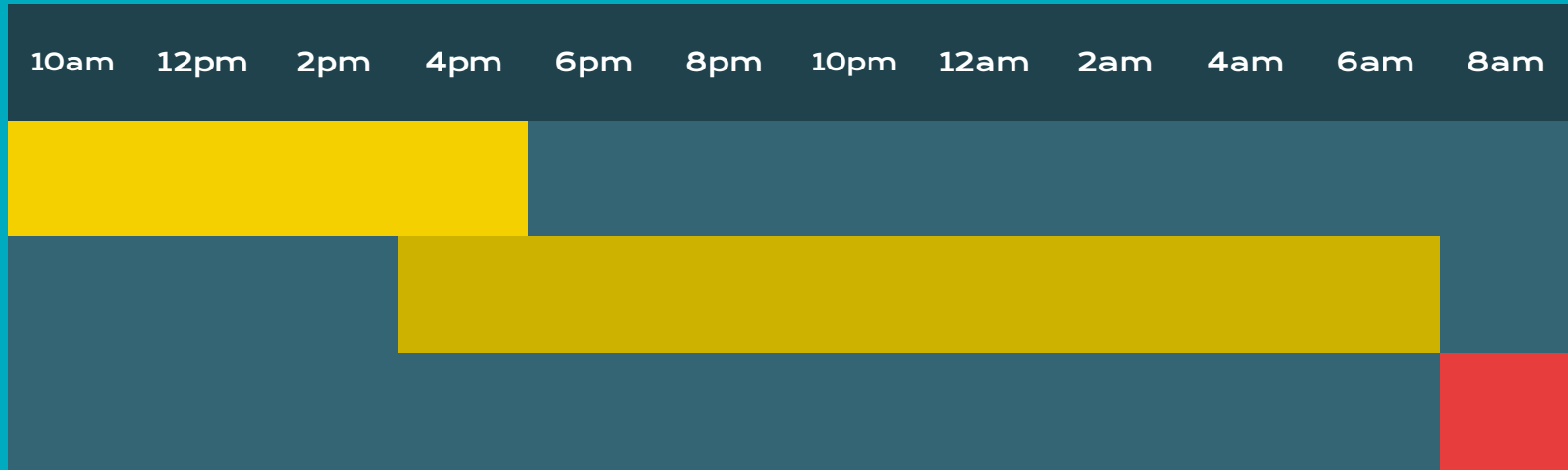
19:48

07-09-2022

Future Layout



Workflow in the 24 hours spent



Planning

Analysing the problem and figuring out the tech stack to be used



Construction

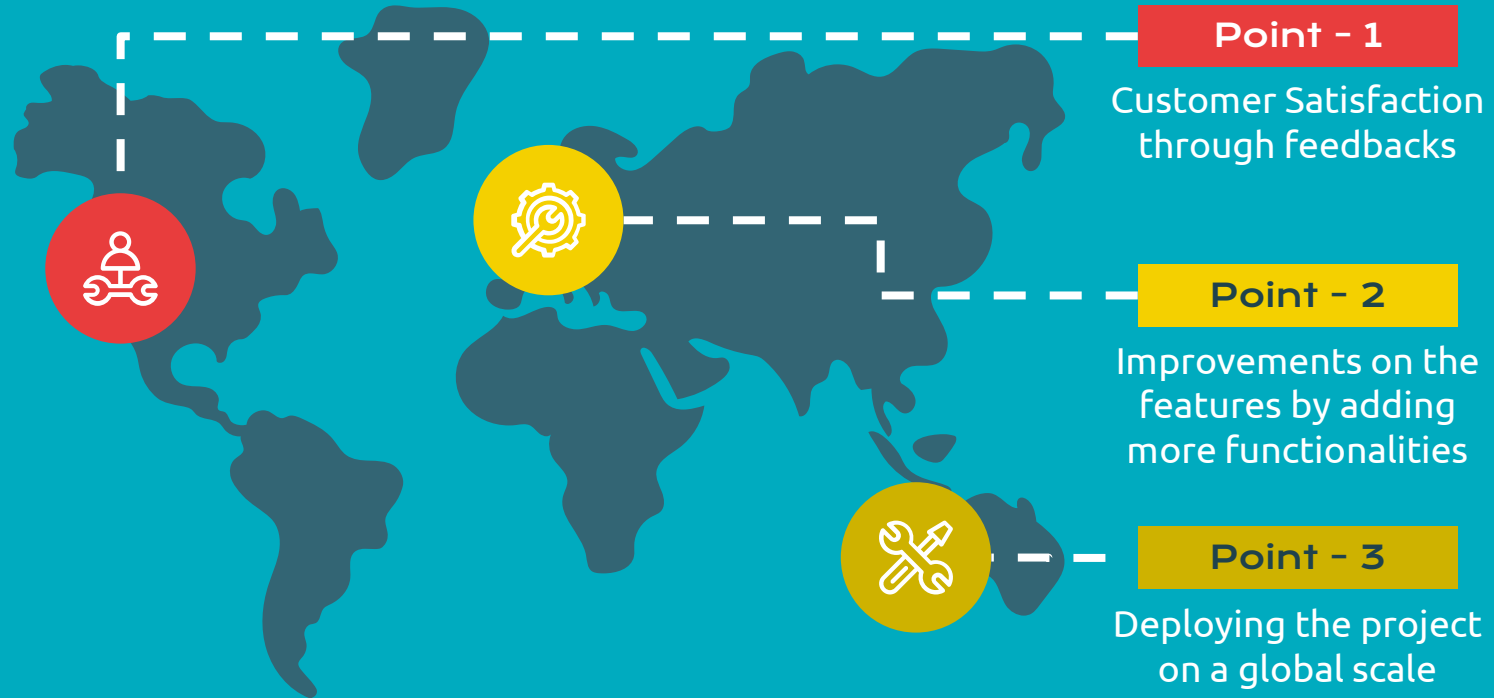
Using TDD, developing the frontend and backend in Django framework



Verification Testing

Debugging and final testing for production on netlify/github pages.

Future Scope



Team Members

SHATAKSHI SHREE

20BAI1314



NAYAN KHEMKA

20BCE1884

DHRUVI OCHANI

20BCE1882



P.V. ABHIRAM

20BAI1132