**README :**

**Overview and Approach to the problem :**

ReportStolenCars is a website where users can provide details about their stolen cars. Police officers are assigned to each case which is registered on the website. Users can track the status of their cases on the site. New cases are automatically assigned to an unoccupied officer, and 1 officer can attend to only 1 case.

**Key approaches to mapping the cases to the officers:**

1) If a case is registered on the site and an officer is free, the case is assigned to the officer and he cannot take on any new cases until the current one is resolved.

2) If there are x cases and x occupied officers the x+1th case is tagged unassigned until a case gets resolved or an officer is added to the force, in which case the x+1th case is assigned to the officer.

3) If there are multiple unassigned cases and 1 unoccupied officer, the earliest registered case is assigned to the officer.

4) If there are multiple unoccupied officers and 1 unassigned case, the case is assigned to the officer which was least recently occupied.

**Technologies used:**

1) Flask, JavaScript (front end)

2) Golang, Gorilla mux (framework like flask) backend.

3) MongoDB database.

**Steps to Run the application :**

1. Install MongoDB
2. Download python 3.5.x or above for your machine.
3. Unzip the file in the directory of your choice
4. Run all the following commands from the root of the unzipped directory.
5. Run ‘pip install -r requirements.txt’.
6. Run ‘python api.py’ which runs on localhost:3000
7. Run the ‘app.exe’ binary, if the machine is not windows or you wish to build the binary in your system, follow the steps below.
8. Install Golang 1.12.x+ on your system.
9. If your machine is windows, set all the path variables correctly (GOPATH and GOROOT).
10. Run ‘go get github.com/gorilla/mux’
11. Run ‘go get go.mongodb.org/mongo-driver’ , ignore the go warning
12. Run ‘go build -o <file-name>’
13. Run ‘start <file-name>.exe’ for windows and ‘./<file-name>’ for Linux. The application runs on port 8080.
14. Type localhost:3000 in your browser URL and the website is ready for exploration

**APIs endpoints at** [**http://localhost:8080**](http://localhost:8080) **:**

**1) /report\_stolen**

**I/p:** image of car, number\_plate, name, brand, model, date\_stolen, location\_stolen

**Functionality:** Write case to MongoDB, check if cop is free, assign case to cop if free,

else tag as unassigned case, in the case of invalid input return bad request

**Other api calls:** /assign\_case

**O/p:** case\_id, tell user to check case status in "Track a case"

**Method:** POST

**Status codes:** 201, 400, 500

**2) /get\_free\_cop**

**I/p:** None

**Functionality:** Query MongoDB for all the cops who are unassigned to cases,

pick the cop who was occupied least recently based on timestamp, return that cop's details

**Other api calls:** None

**O/p:** Free cop who was least recently occupied in a case, if no cop is free return bad request

**Method:** GET

**Status codes:** 200, 400, 500

**3) /assign\_case**

**I/p:** None

**Functionality:** Search for a free cop, if no free cop is present return bad request,

search for an unassigned case, if no unassigned case is present return bad request,

if there is a free cop and an unassigned case assign the cop that case and update both

the cop and case tables in MongoDB.

**Other api calls:** /get\_free\_cop,/get\_unassigned\_case

**O/p:** None

**Method:** POST

**Status codes:** 200, 400, 500

**4) /resolve\_case**

**I/p:** caseID

**Functionality:** Free the cop which was assigned to case, tag the case as resolved, search

for an unassigned case since a cop is now available, if the case id is invalid or the case is unassigned,

return bad request

**Other api calls:** /free\_cop,/tag\_as\_resolved,/assign\_case

**Method:** POST

**O/p:** None

**Status codes:** 200, 400, 500

**5) /free\_cop**

**I/p:** caseID

**Functionality**: Free the cop which was assigned to the case in question, return bad request in

case of an invalid case or case which is unassigned

**Other api calls:** None

**Method:** POST

**O/p:** None

**Status codes:** 200, 400, 500

**6) /get\_unassigned\_case**

**I/p:** None

**Functionality:** Search for all cases which are tagged as unassigned and return the case which

was registered the earliest in the website, if no cases are unassigned return bad request.

**Other api calls:** None

**O/p:** case details of the earliest registered case from the DB.

Method: GET

**Status codes:** 200, 400, 500

**7) /track\_case**

**I/p:** caseID

**Functionality:** Return all the case details from the DB, return bad request and error message if

invalid caseID

**Other api calls:** None

**O/p:** caseID reported\_by handled\_by status image number\_plate date\_stolen date\_recovered (if applicable)

**Method:** GET

**Status codes:** 200, 400, 500

**8) /tag\_as\_resolved**

**I/p:** caseID

**Functionality:** Update the case as resolved in the case Table in MongoDB, return bad request if invalid case id.

**Other api calls:** None

**O/p:** None

**Method:** POST

**Status codes:** 200, 400, 500

**9) /add\_cop**

**I/p:** name

**Functionality:** add cop to database, return his details, look for unassigned cases for the new cop. Return bad request for invalid input.

**Other api calls:** /assign\_case

**O/p:** name of cop, status, no\_of\_cases

**Method:** POST

**Status codes:** 200, 400, 500

**Following are the name of the Tables and their schemas in MongoDB:**

**List of cops**

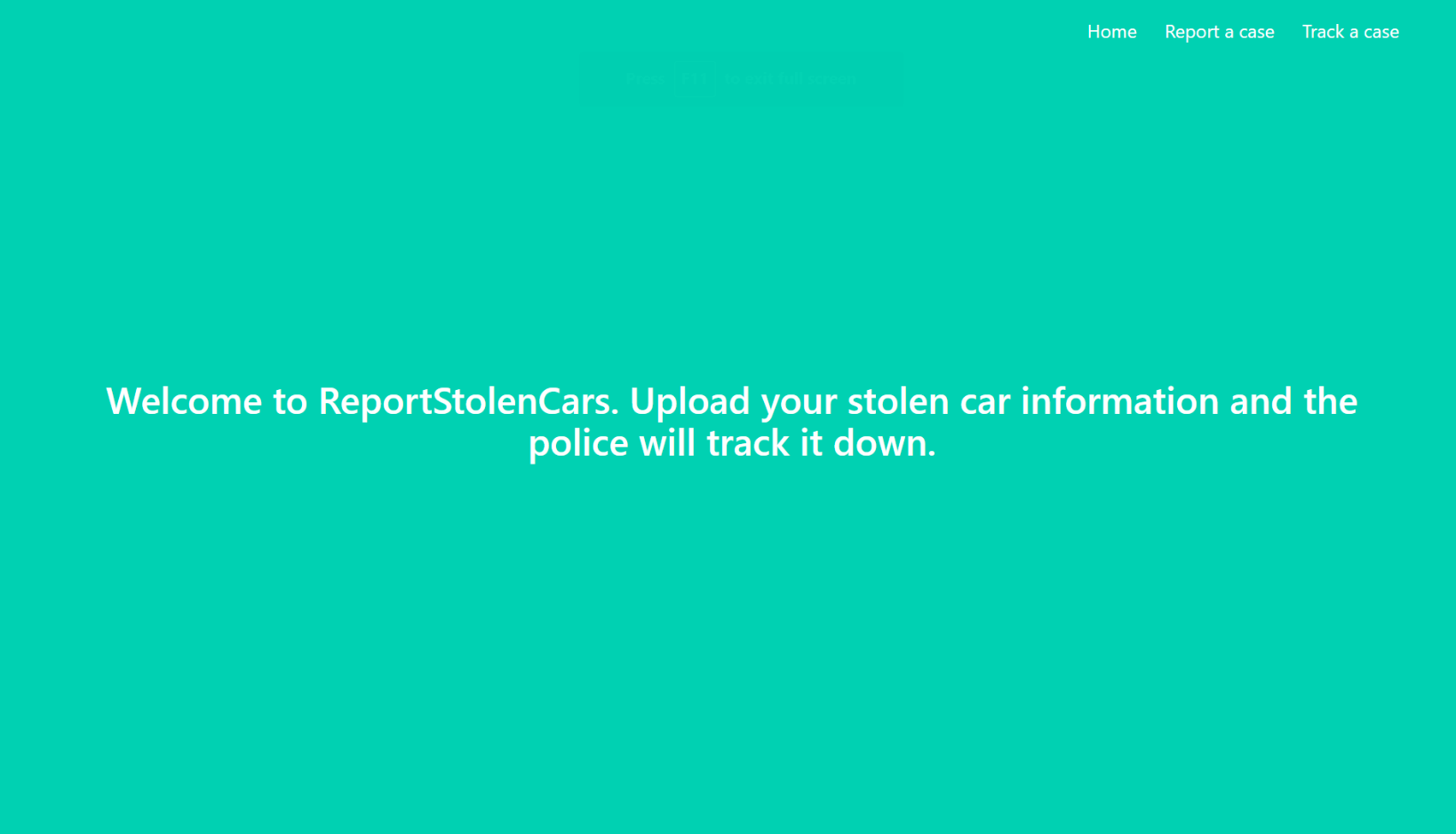
id name no\_of\_cases status

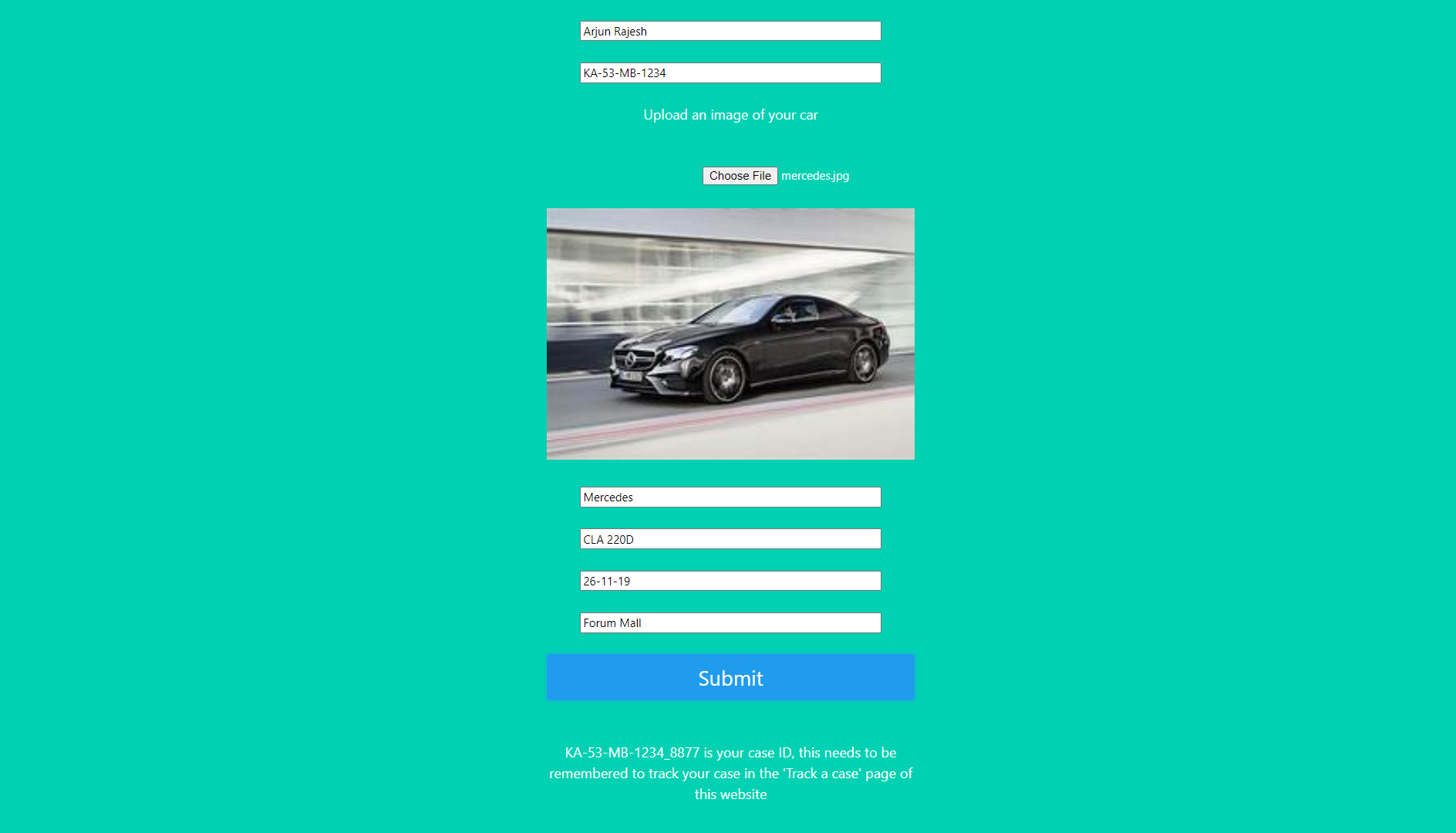
**List of cases**

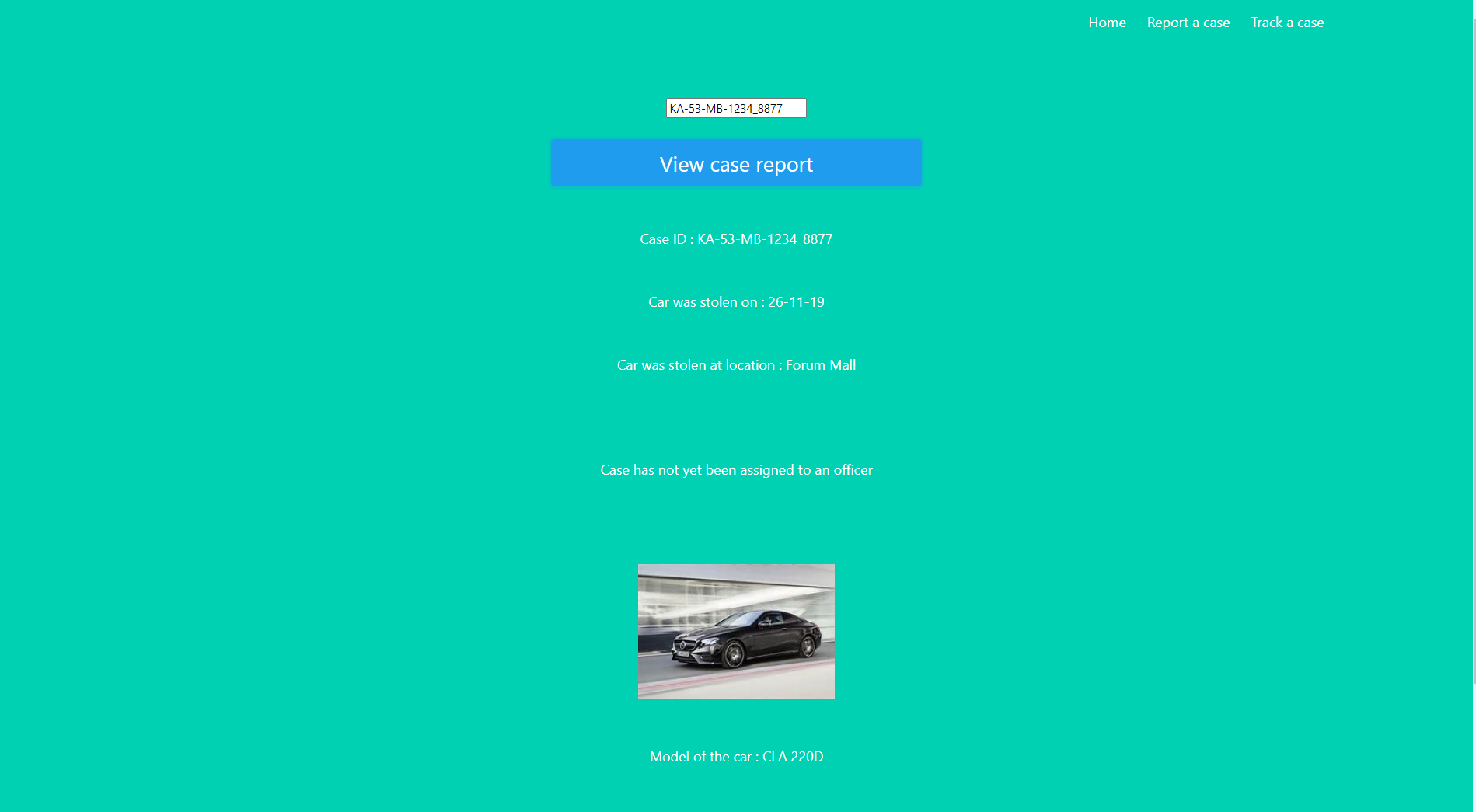
id case\_id reported\_by handled\_by status image number\_plate location\_stolen date\_stolen date\_recovered model brand timestamp

**Sample Screenshots:**

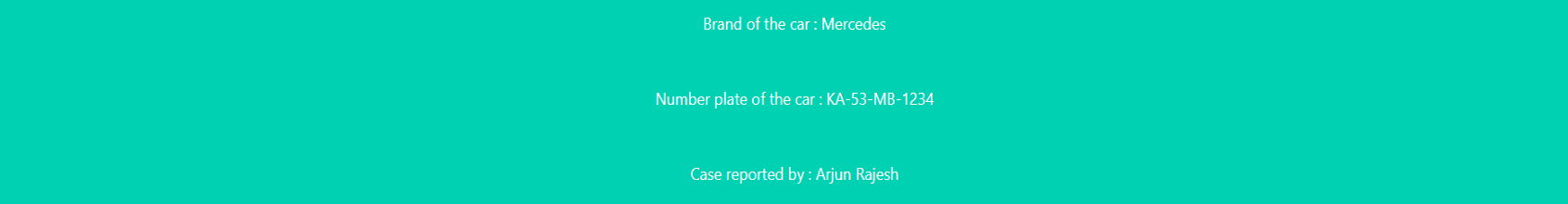
**The following are the situations displayed in the screenshots:**

1. The enters the home page
2. The user enters his car details and receives a case ID to remember.

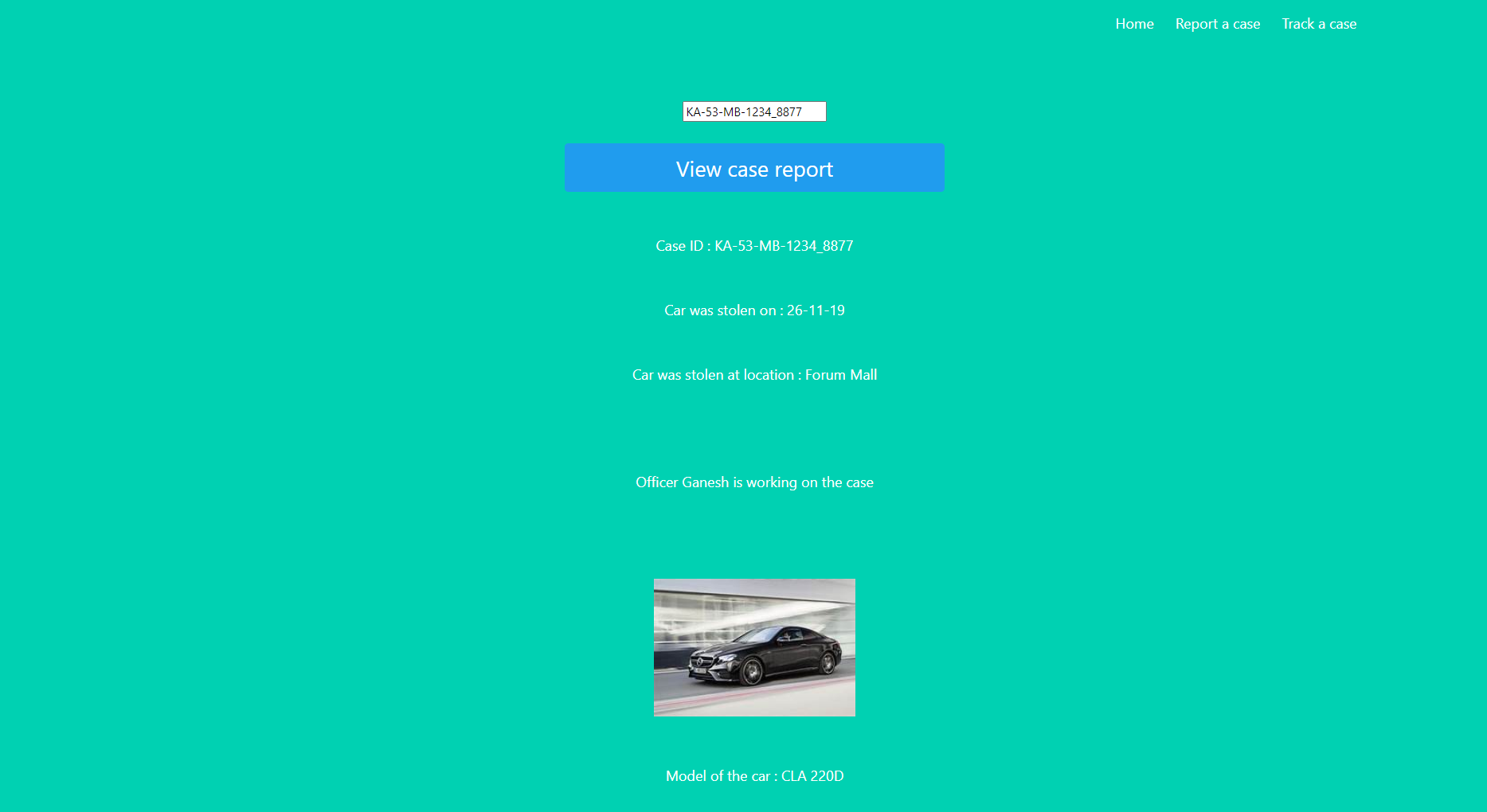


1. The user enters the track a case button and notices that the case has not been assigned to an officer yet

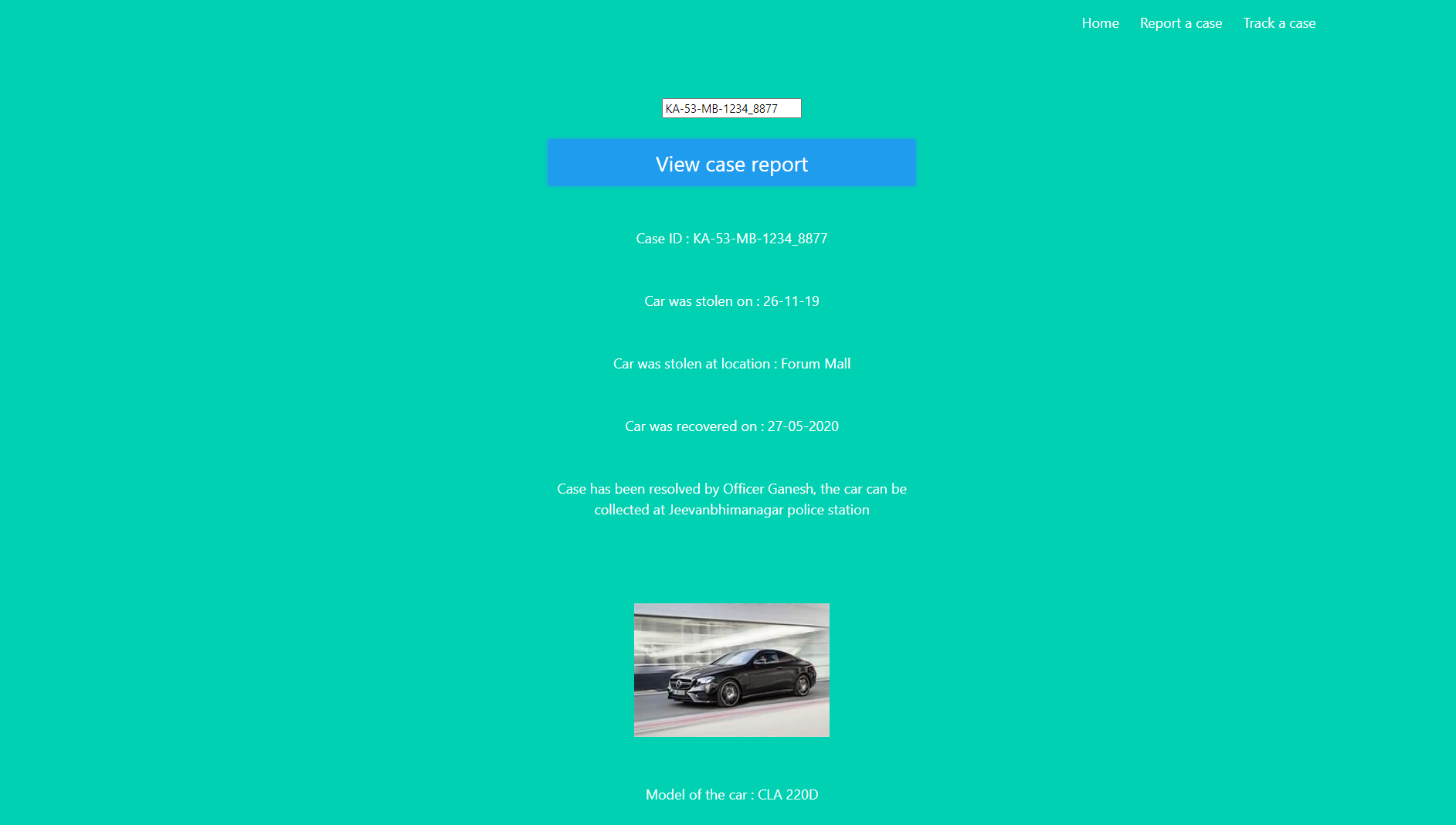
The image below is a continuation of the web page in the screenshot above and these fields will be skipped for the subsequent screenshots as they are redundant.



1. At a later attempt to track the same case the user notices an officer becomes unoccupied as some other case gets resolved and he/she gets assigned to the case.



1. The user later checks again and notices that the case has been resolved on 27-05-2020 and the vehicle can be picked up at the police station.



1. The user later enters a dummy Case ID to see what happens and gets an invalid case ID error message.

