ANALYZE InfoTech

2022

SUMMER

TRAINING

REPORT

SASHANK SHEKHAR SHUKLA

2000143250l3

MCA-II

SUMMER TRAINING REPORT

ON

Web development

using Python and Django

[BATCH – 2020 TO 2022]

[MCA II-YEAR III-SEMESTER]

SUBMITTED TO:

ER. PANKAJ ROY

ASSIST. PROF. FoET

UNIV. OF LUCKNOW

SUBMITTED BY:

SASHANK SHEKHAR SHUKLA [200014325013]

MCA III-SEMESTER

Web development

using Python and Django

From

Under the guidance of

Mr. Ankit Singh

(Subject Matter Expert)

Report prepared by

SASHANK SHEKHAR

MCA-III

CERTIFICATE

Analyze Infotech, Janpriya Complex Phase-II,
Tedhi Pulia, Aliganj, Lucknow,(U.P.)
Phone :-9453193268, 9839434821,
Web :-www.analyzeinfotech.in
E-mail :-admin@analyzeinfotech.in

Ref. No. ai/21/674

TO WHOM SO EVER IT MAY CONCERN

This is to certify that **Mr. Sashank Shekhar Shukla** S/O Mr. Bal Krishna Shukla student of MCA from University of Lucknow, Lucknow in recognition of successful completion project of "**Courier Tracking**" on Python with Django as Front End, SQLite as Back End under the guidance of Mr. Ankit Singh (Project Manager) at Analyze InfoTech. The duration of his project was from 22 July 2021 to 25 September 2021.

He has done a good job during his engagement with the software Development & Testing Division of Company. He has completed his project during the Internship tenure. His performance was good and satisfactory.

I would like to take this opportunity to express my appreciation to Mr. **Sashank Shekhar Shukla** for his work and wish him All the best for future endeavors.

Yours Sincerely

Date: 07 January 2022

Pankaj Mishra

(Project Manager)

For Analyze InfoTech, Lucknow.

Acknowledgement

I offer sincere thanks and humble regards to **ANALYZE TRAINING CENTER**, based in Lucknow, U.P., India for imparting us very valuable professional training in MCA.

I pay my gratitude and sincere thanks to **Mr. Ankit Singh**, my subject matter expert for giving me the cream of his knowledge, I am grateful to him as he has been a constant source of advice, motivation and inspiration.

I am also thankful to him for giving his suggestions and encouragement throughout the project work.

I take the opportunity to express gratitude and thanks to our computer labs staff for providing me opportunity to utilize the available resources for the completion of the project.

Finally, my thanks and appreciation go to my collage's young and dynamic teacher, Mr. Pankaj Roy, who irrespective of the situation, always encouraged and supported me to prepare this project report.

SASHANK SHEKHAR SHUKLA

[200014325013]

INDEX

LIST OF FIGURES

1 INTRODUCTION

What is a web application?

A web application is a computer program that uses a web browser to perform a particular function. It is also called a web app. Web apps are present on many websites. A simple example is a contact form on a website.

A web application is a client-server program. It means that it has a client-side and a server-side. The term "client" here refers to the program the individual uses to run the application. It is part of the client-server environment, where many computers share information. For example, in the case of a database, the client is the program through which the user enters data. The server is the application that stores the information.

Businesses need to exchange information and conclude transactions with their target customers. The Internet can be an excellent and inexpensive channel for that purpose, providing that there is a way to capture and store all the necessary data and show results to users. Thanks to web applications, users can interact with the business using shopping carts or content management systems.

Web apps can be developed for many different reasons and used by companies or individuals. Individuals need it to facilitate their communication or purchase things online. Also, employees can collaborate on projects and work on shared documents with web applications. They can create reports, files, and share information from anywhere and with any device.

Web apps have evolved since their invention. One of the first applications, Perl, a popular server-side scripting language, was developed in 1987. That was before the Internet really became popular outside academic and technology circles. The first web applications were relatively simple and became more sophisticated in the late '90s. Today, they are part of the everyday lives of millions of Americans.

e.g.,



Python Programming Language

Python is an easy to learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming. Python's elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms.

The Python interpreter and the extensive standard library are freely available in source or binary form for all major platforms from the Python web site, https://www.python.org/, and may be freely distributed. The same site also contains distributions of and pointers to many free third party Python modules, programs and tools, and additional documentation.

The Python interpreter is easily extended with new functions and data types implemented in C or C++ (or other languages callable from C). Python is also suitable as an extension language for customizable applications.

About Django Framework

Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source.

Ridiculously fast:

Django was designed to help developers take applications from concept to completion as quickly as possible.

Reassuringly secure:

Django takes security seriously and helps developers avoid many common security mistakes.

Exceedingly scalable:

Some of the busiest sites on the web leverage Django's ability to quickly and flexibly scale.

Training Objectives:

1. Understand the principles of creating an effective web page, including an in-depth consideration of information architecture.
2. Become familiar with graphic design principles that relate to web design and learn how to implement theories into practice.
3. Develop skills in analysing the usability of a web site.
4. Understand how to plan and conduct user research related to web usability.
5. Learn the language of the web: HTML, CSS and JavaScript .
6. Learn techniques of responsive web design, including media queries.
7. Develop skills in web security.
8. Develop basic programming skills using Python.
9. Be able to embed dynamic content into web pages and secure transport of the data.

2 ABOUT ORGANISATION

Analyze Infotech is a rapidly growing IT software company specialising in software technologies of recent trends such as Java, .Net, Android, PHP, Web Designing, WordPress, Joomla, and Search Engine Optimization & Social Media Marketing. Analyze Infotech is a professionally operated foundation imparting full range of understanding in the software technology that is considered to be mandatory in the contemporary IT scenario.

At Analyze Infotech they provide complete end-to-end solutions based on our expertise in the field of e-technology. They undertake projects involving technologies like J2EE, JSP, ASP.NET, PHP, CMS(WordPress, Joomla, Magento, Drupal), Flash, Director, 3D-Max, XML and RDBMS, such as Oracle and SQL server. From helping companies integrate customer-centric strategies and emerging technologies into innovative e-business models, to developing multi-channel solutions that coordinate these new channels with the traditional customer touch points of telephone, catalogues, call centres and brick-and-mortar facilities.

They offer best Python and Java Training in Lucknow. They have expert trainers with real time experience and having excellent teaching track. Their training course ensure that students learn the core and advanced topics required for a successful career in the field of Java Development.

Their training course has helped many job seekers, corporate professionals and learners to improve their technical as well as soft skills in order to become a proficient professional.

Academy of Learning offers students:

1. Deliver an effective approach to career training.
2. Provide industry standard equipment and software.
3. Offer career specific programs.
4. Maintain an environment conducive to learning.
5. Schedule flexible class hours.
6. Offer job search assistance and services.

3 DESCRIPTION OF TECHNOLOGY

Technology, the application of scientific knowledge to the practical aims of human life or, as it is sometimes phrased, to the change and manipulation of the human environment.

Software Requirements:

Visual Studio Code

Visual Studio Code is a lightweight but powerful source code editor which runs on your desktop and is available for Windows, macOS and Linux. It comes with built-in support for JavaScript, TypeScript and Node.js and has a rich ecosystem of extensions for other languages (such as C++, C#, Java, Python, PHP, Go) and runtimes (such as .NET and Unity).

Visual Studio Code supports Portable mode installation. This mode enables all data created and maintained by VS Code to live near itself, so it can be moved around across environments, for example, on a USB drive.

VS Code is an editor, first and foremost, and prides itself on a small footprint. Unlike traditional IDEs that tend to include everything but the kitchen sink, you can tune your installation to the development technologies you care about.

Python

1. Python is a programming language that lets you work quickly and integrate systems more effectively.
2. Python is free and easy to learn.
3. Python is an interpreted high-level general-purpose programming language.
4. Its design philosophy emphasizes code readability with its use of significant indentation.
5. Its language constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Chrome

Make the most of the Web with impeccably optimized, personalized, synced, and secured browsing.

Google's Chrome Web browser has become one of the most popular in the world, thanks to smooth performance, support for add-ons, and features like casting and voice search that are absent in or only partially implemented by competing browsers like Safari, Mozilla Firefox, and Microsoft Edge.

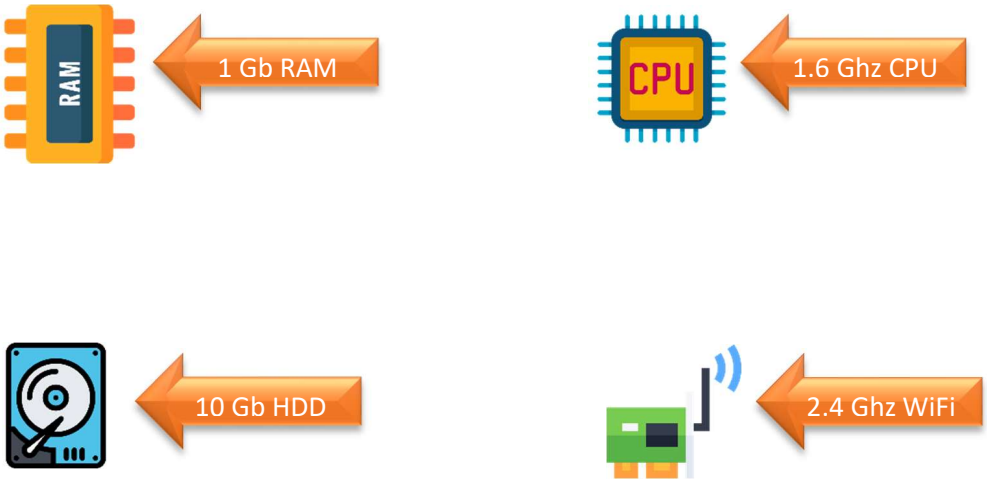
The best add-on support: Chrome slightly edges out Firefox in two ways.

Chrome's task manager (access it by pressing Shift-Esc) breaks down how much RAM and CPU power each add-ons is using, so you can identify ones that may be causing issues with browser performance or device battery life.

Django Framework

1. Django is a web application framework written in Python programming language.
2. It is based on MVT (Model View Template) design pattern.
3. The Django is very demanding due to its rapid development feature.
4. It takes less time to build application after collecting client requirement.
5. This framework uses a famous tag line:The web framework for perfectionists with deadlines.
6. By using Django, we can build web applications in very less time.
7. Django is designed in such a manner that it handles much of configure things automatically, so we can focus on application development only.

Minimum Hardware Requirements:



4 DESCRIPTION OF PROJECT

PROJECT NAME → COURIER TRACKING

OBJECTIVE → Applying the knowledge of Python to build a web project “ Courier Tracking ”.

Header:

Logo

Logos are images, texts, shapes, or a combination of the three that depict the name and purpose of a business – to put it simply.

However, a logo can and should be more than a symbol of identification.

If designed well, it also tells a company's story, by conveying your brand message in a way that helps to establish an emotional connection with your target audience.

A logo is important for a number of reasons, mainly being that it:

1. Makes a great first impression, which invites customers to interact with your brand.
2. Helps you to create a brand identity.
3. Gives your company a symbol through which people can better remember you.
4. Distinguishes you from competitors.
5. Fosters brand loyalty.

Figure 1. LOGO

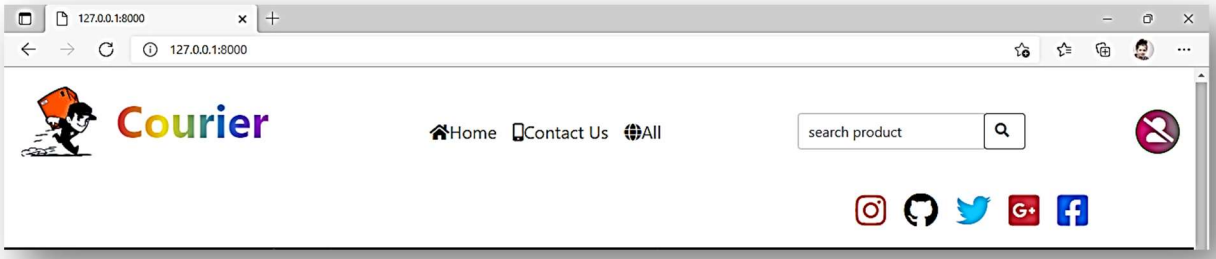


Figure 2. Header when no user is logged-in

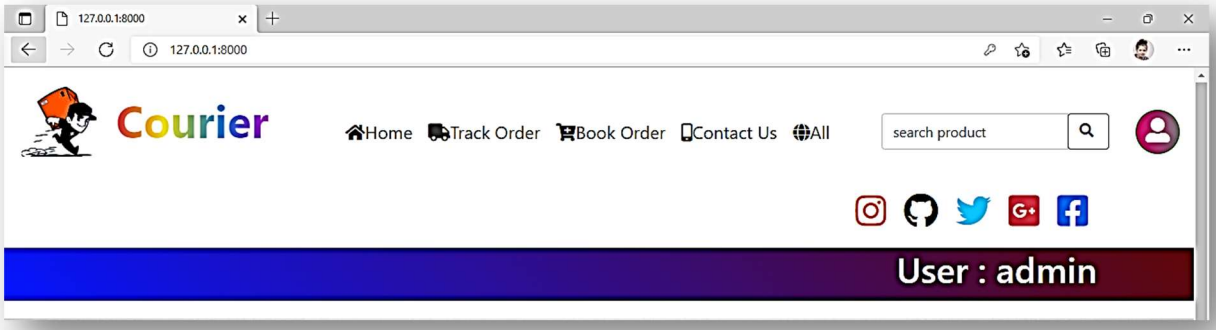


Figure 3. Header when user is logged-in

Menu

When creating a website, the different elements of the page should come together in guiding visitors through your site seamlessly and with ease. One element that plays a decisive role in the user experience of your site and greatly affects navigation, is the menu.

What are website menus?

A website menu is a series of linked items that serve in navigating between the different pages or sections of a website. There are several kinds of menus, depending on the website's content and design. The main types of website menus are:

1. **Classic navigation menu:** This most widespread kind of menu is placed in the website's header, typically as a horizontal list.
2. **Sticky menu:** Also known as a fixed or floating menu, this menu stays put as visitors scroll down the site. These are ideal for long-scrolling pages.
3. **Hamburger menu:** An icon made up of three horizontal stripes that opens up into a menu once clicked. This design convention is rooted in mobile website design, but is widely used on desktop as well.
4. **Dropdown menu:** A menu in which a list of additional items opens up once visitors click on - or hover over - one of the menu items. This option is suitable for websites with a lot of content.
5. **Sidebar menu:** A list of menu items that's located on the left or right side of a webpage.

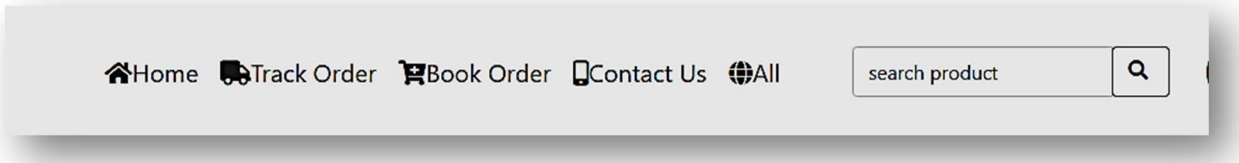


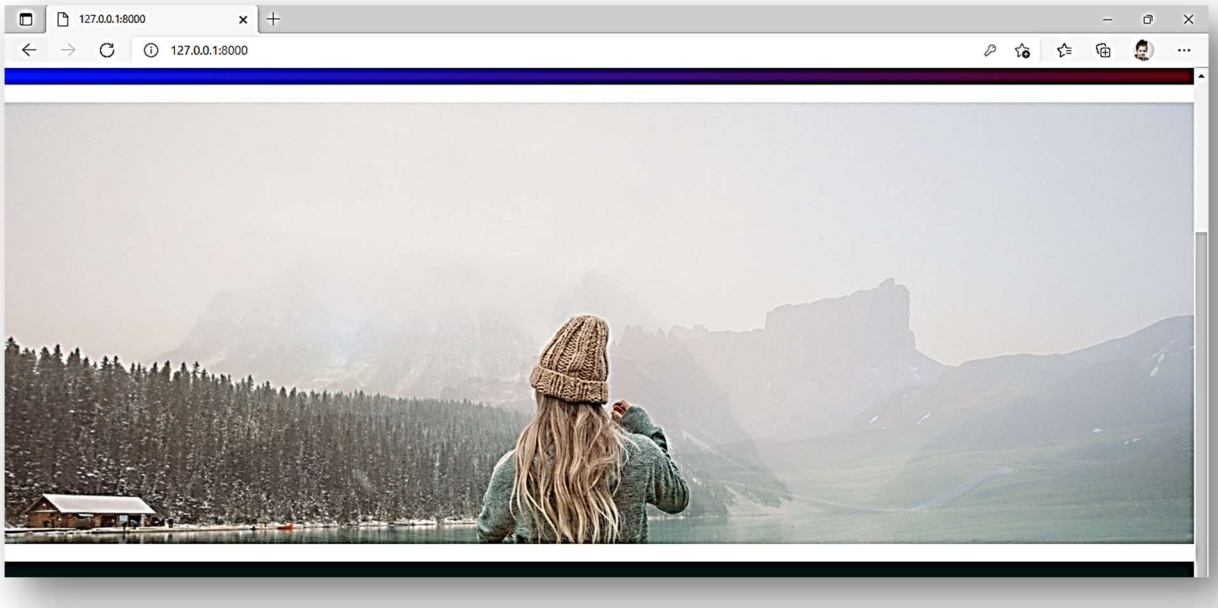
Figure 4. Menu

Body:

Interaction

CSS is the language we use to style an HTML document.

CSS describes how HTML elements should be displayed.



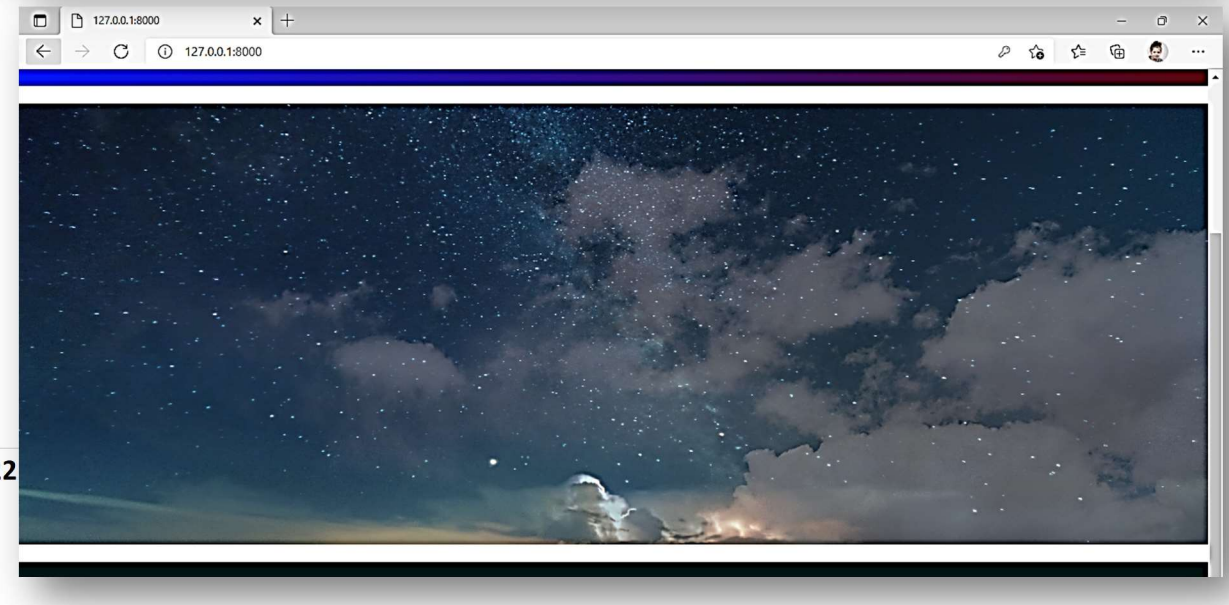


Figure 5.Body

Forms

An HTML form is used to collect user input. The user input is most often sent to a server for processing.

The action attribute defines the action to be performed when the form is submitted.

The target attribute specifies where to display the response that is received after submitting the form.

The target attribute can have one of the following values:

Value	Description
_blank	The response is displayed in a new window or tab
_self	The response is displayed in the current window
_parent	The response is displayed in the parent frame
_top	The response is displayed in the full body of the window

framename	The response is displayed in a named iframe

1. The method attribute specifies the HTTP method to be used when submitting the form data.
2. The form-data can be sent as URL variables (with method="get") or as HTTP post transaction (with method="post").
3. The default HTTP method when submitting form data is GET.

1. Appends the form data to the URL, in name/value pairs
2. NEVER use GET to send sensitive data! (the submitted form data is visible in the URL!)
3. The length of a URL is limited (2048 characters)
4. Useful for form submissions where a user wants to bookmark the result
5. GET is good for non-secure data, like query strings in Google

1. Appends the form data inside the body of the HTTP request (the submitted form data is not shown in the URL)
2. POST has no size limitations, and can be used to send large amounts of data.
3. Form submissions with POST cannot be bookmarked

The HTML <form> element can contain one or more of the following form elements:

<input> : Defines input field

<label> : Describes input field

<select> : Provide selection mechanism

<textarea> : For a multiline text, takes rows and columns in numbers

<button> : For clickable buttons

<fieldset> : Wraps the elements

<legend> : Names the fieldset

<datalist> : Provide some frequent options

<output> :

<option> : For choices in datalist/select

<optgroup>

Input

<input type="button"> : Clickable button

<input type="checkbox"> : Multiple choices

<input type="color"> : Choosing color

<input type="date"> : For date

<input type="datetime-local"> : Local date and time

<input type="email"> : Insures that input field have email

<input type="file"> : For file

<input type="hidden"> : Hides the option

<input type="image"> : Image files

<input type="month">

<input type="number"> : Numbers only

<input type="password"> : Hides input data

<input type="radio"> : One out of many

<input type="range"> : Specifies range for sliders

<input type="reset"> : Clears the form if filled any

<input type="search"> : defines a text field for entering a search string

<input type="submit"> : Final submission button

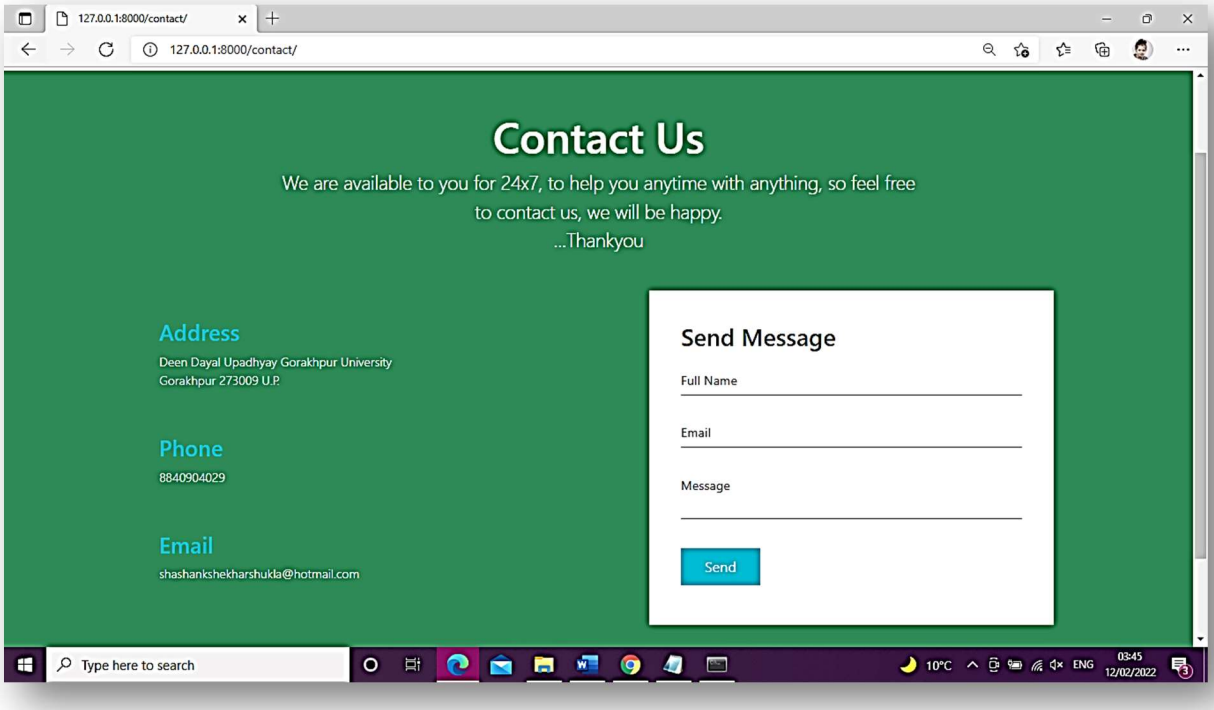
<input type="tel"> : Telephone number

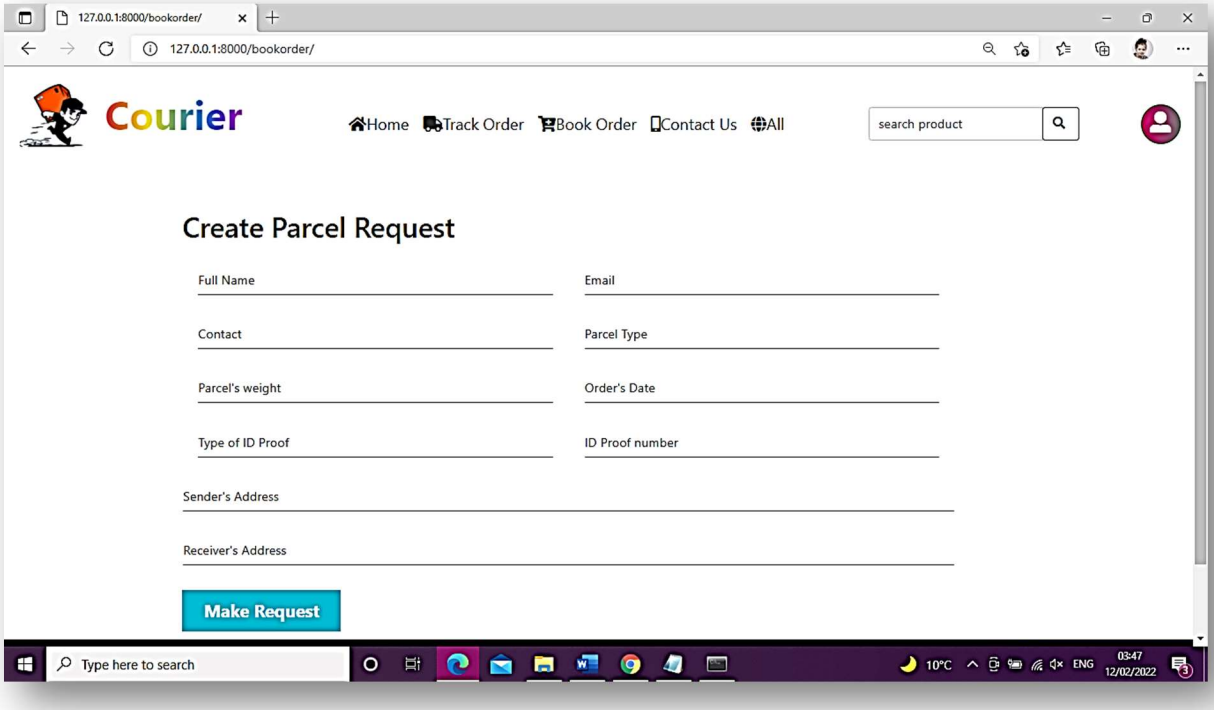
<input type="text"> : Single line text

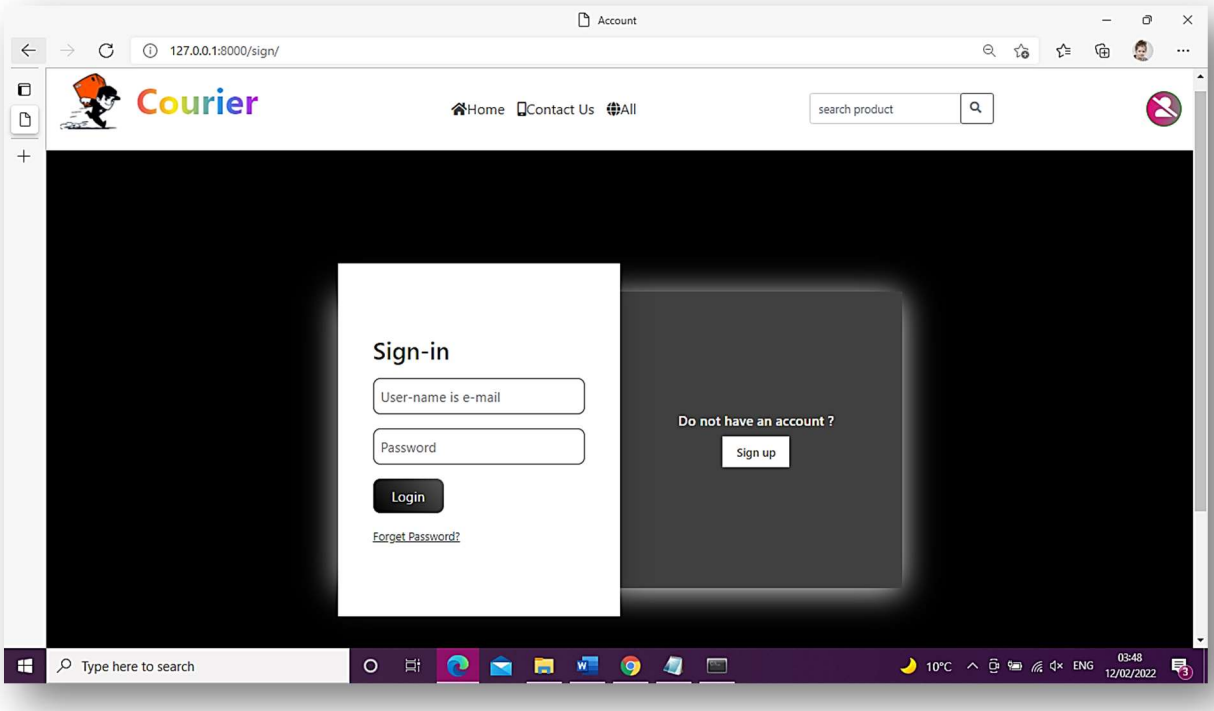
<input type="time"> : Time

<input type="url"> : Accepts valid urls only

<input type="week"> : Time duration in form of Weeks




127.0.0.1:8000/bookorder/
127.0.0.1:8000/bookorder/
Courier
Home
Track Order
Book Order
Contact Us
All
search product
Create Parcel Request
Full Name
Email
Contact
Parcel Type
Parcel's weight
Order's Date
Type of ID Proof
ID Proof number
Sender's Address
Receiver's Address
Make Request
Type here to search
10°C
ENG
03:47
12/02/2022


Account
127.0.0.1:8000/sign/
Courier
Home
Contact Us
All
search product
Sign-in
User-name is e-mail
Password
Login
Forget Password?
Do not have an account ?
Sign up
Type here to search
10°C
ENG
03:48
12/02/2022

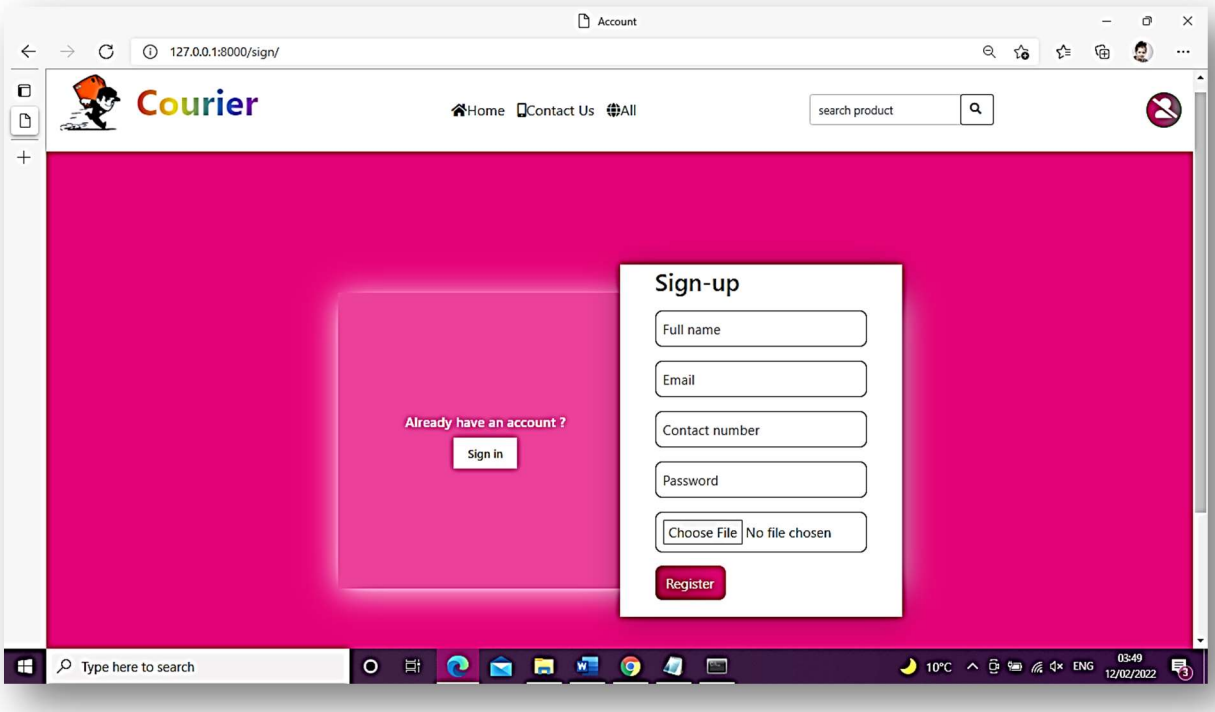


Figure 6. Forms in the web-application

NOTES

1. The input value attribute specifies an initial value for an input field.
2. The input readonly attribute specifies that an input field is read-only.
3. The input disabled attribute specifies that an input field should be disabled.
4. A disabled input field is unusable and un-clickable.
5. The input size attribute specifies the visible width, in characters, of an input field.
6. The default value for size is 20.
7. The input maxlength attribute specifies the maximum number of characters allowed in an input field.
8. The input multiple attribute specifies that the user is allowed to enter more than one value in an input field.
9. The multiple attribute works with the following input types: email, and file.

10. The input placeholder attribute specifies a short hint that describes the expected value of an input field (a sample value or a short description of the expected format).
11. The short hint is displayed in the input field before the user enters a value.
12. The placeholder attribute works with the following input types: text, search, url, tel, email, and password.
13. The input required attribute specifies that an input field must be filled out before submitting the form.
14. The required attribute works with the following input types: text, search, url, tel, email, password, date pickers, number, checkbox, radio, and file.
15. The input height and width attributes specify the height and width of an <input type="image"> element.
16. The input list attribute refers to a <datalist> element that contains pre-defined options for an <input> element.

Footer:

In general, a footer is an area at the bottom of a document page containing data common to other pages. The information in footers may include page numbers, creation dates, copyrights, or references that appears on a single page, or on all pages.

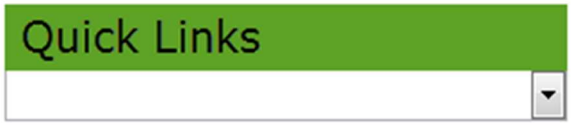
Copyright

Copyright, like other intellectual property rights, is subject to a statutorily determined term. Once the term of a copyright has expired, the formerly copyrighted work enters the public domain and may be used or exploited by anyone without obtaining permission, and normally without payment. However, in paying public domain regimes the user may still have to pay royalties to the state or to an authors' association.

Quick Links

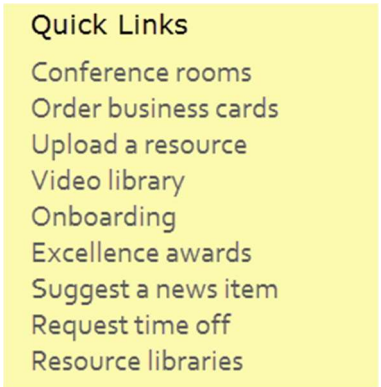
A Quicklinks UI design pattern often surfaces as a poor fix for addressing findability and discoverability issues on intranets. Acting as a catch-all for different types of links causes a separate set of issues. And Quicklinks is always a vague label.

Quicklinks (or Quick Links) is a list of unstructured links placed in a salient place on a page. The feature usually takes the form of a drop-down list appearing in the upper right part of the homepage or on all pages on the intranet.



Often referred to as a "navigator," this type of component has several benefits. Mainly, it resides at the high level on the page as well as in the information architecture can attract attention to a small set of links or commands may make some commands easier to find and access than they are in the IA (Navigators are sometimes also used on intranets to access elements that are not part of the intranet's IA. For example, designers may use a navigator outside the intranet's global navigation to offer access to another enterprise application or the public-facing website. This type of navigator is better justified than Quicklinks.)

While Quicklinks is usually a drop-down list, sometimes it can also be presented in a more visible and accessible way, as an on-page list of links.



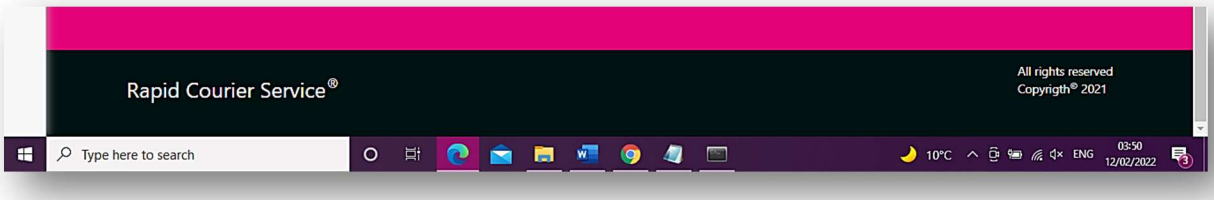


Figure 7. Footer

Authentication:

Django comes with a user authentication system. It handles user accounts, groups, permissions and cookie-based user sessions.

Overview:

The Django authentication system handles both authentication and authorization. Briefly, authentication verifies a user is who they claim to be, and authorization determines what an authenticated user is allowed to do. Here the term authentication is used to refer to both tasks.

The auth system consists of

Users

Permissions: Binary (yes/no) flags designating whether a user may perform a certain task.

Groups: A generic way of applying labels and permissions to more than one user.

A configurable password hashing system

Forms and view tools for logging in users, or restricting content

A pluggable backend system

The authentication system in Django aims to be very generic and doesn't provide some features commonly found in web authentication systems. Solutions for some of these common problems have been implemented in third-party packages:

1. Password strength checking
2. Throttling of login attempts
3. Authentication against third-parties (OAuth, for example)
4. Object-level permissions

Installation

Authentication support is bundled as a Django contrib module in django.contrib.auth. By default, the required configuration is already included in the settings.py generated by django-admin startproject, these consist of two items listed in your INSTALLED_APPS setting:

'django.contrib.auth' contains the core of the authentication framework, and its default models.

'django.contrib.contenttypes' is the Django content type system, which allows permissions to be associated with models you create.

and these items in your MIDDLEWARE setting:

SessionMiddleware manages sessions across requests.

AuthenticationMiddleware associates users with requests using sessions.

With these settings in place, running the command manage.py migrate creates the necessary database tables for auth related models and permissions for any models defined in your installed apps.

1. Using Django's default implementation
2. Working with User objects
3. Permissions and authorization
4. Authentication in web requests
5. Managing users in the admin
6. API reference for the default implementation

Password management in Django

User objects

User objects are the core of the authentication system. They typically represent the people interacting with your site and are used to enable things like restricting access, registering user profiles, associating content with creators etc. Only one class of user exists in Django's authentication framework, i.e., 'superusers' or admin 'staff' users are just user objects with special attributes set, not different classes of user objects.

The necessary attributes of the default user are:

1. username
2. password
3. email
4. first_name
5. last_name

Creating usres

The most direct way to create users is to use the included create_user() helper function:

```
>>> from django.contrib.auth.models import User
>>> user = User.objects.create_user('john', 'lennon@thebeatles.com', 'johnpassword')

# At this point, user is a User object that has already been saved
```

```
# to the database. You can continue to change its attributes
# if you want to change other fields.
>>> user.last_name = 'Lennon'
>>> user.save()
```

Admin authorization:

Create superusers using the createsuperuser command:

```
$python manage.py createsuperuser --username=sss --email=sss@hotmail.com
```

You will be prompted for a password. After you enter one, the user will be created immediately. If you leave off the --username or --email options, it will prompt you for those values.

Django comes with a built-in permissions system. It provides a way to assign permissions to specific users and groups of users.

It's used by the Django admin site, but you're welcome to use it in your own code.

The Django admin site uses permissions as follows:

1. Access to view objects is limited to users with the "view" or "change" permission for that type of object.

2. Access to view the “add” form and add an object is limited to users with the “add” permission for that type of object.
3. Access to view the change list, view the “change” form and change an object is limited to users with the “change” permission for that type of object.
4. Access to delete an object is limited to users with the “delete” permission for that type of object.

Data Communication:

Data Communication is a deciding factor for the performance of the website.

If a website is designed in such a way that it communicates very efficiently with the server, then it will take less bandwidth to work properly, and if not designed so, it will take time to load and may end user’s interest in the web service.

So, a good developer always keeps it in mind and sends or retrieves data what is necessary to be fetched from server as minimum as possible without affecting the quality and finds a solution to problem will increase unnecessary load on the server or just removes it.

Me as being developer of this project, coded the website in such a way that it communicates only necessary data to the server best to my knowledge.

I saw during testing that for gallery, the website is requesting photos continuously from the server, which I found very bad, so I used **Image Sprite** to pack images into one and used CSS animation to display them one by one.

Overall, the minimal communication can be achieved by good knowledge of the technology and its use, like optimised code at Server side and more text less multimedia at client side [Not in all cases, e.g., Multimedia is the key for Gaming, Streaming and Video hosting websites]

Form Data

All the forms used in this project uses POST method to deliver data to the Server to ensures the integrity of data.

Forms are designed after thorough study of processes, and only sensible data is kept with proper security and format, others are discarded from the Form.

Forms are used only when necessary and does not ask for any such data which may harm user in present or future.

CSS is used to beautify forms and JavaScript is used for client-side validation of the data.

Separate functions are used for forms to avoid conflict and for fast response to the end user.

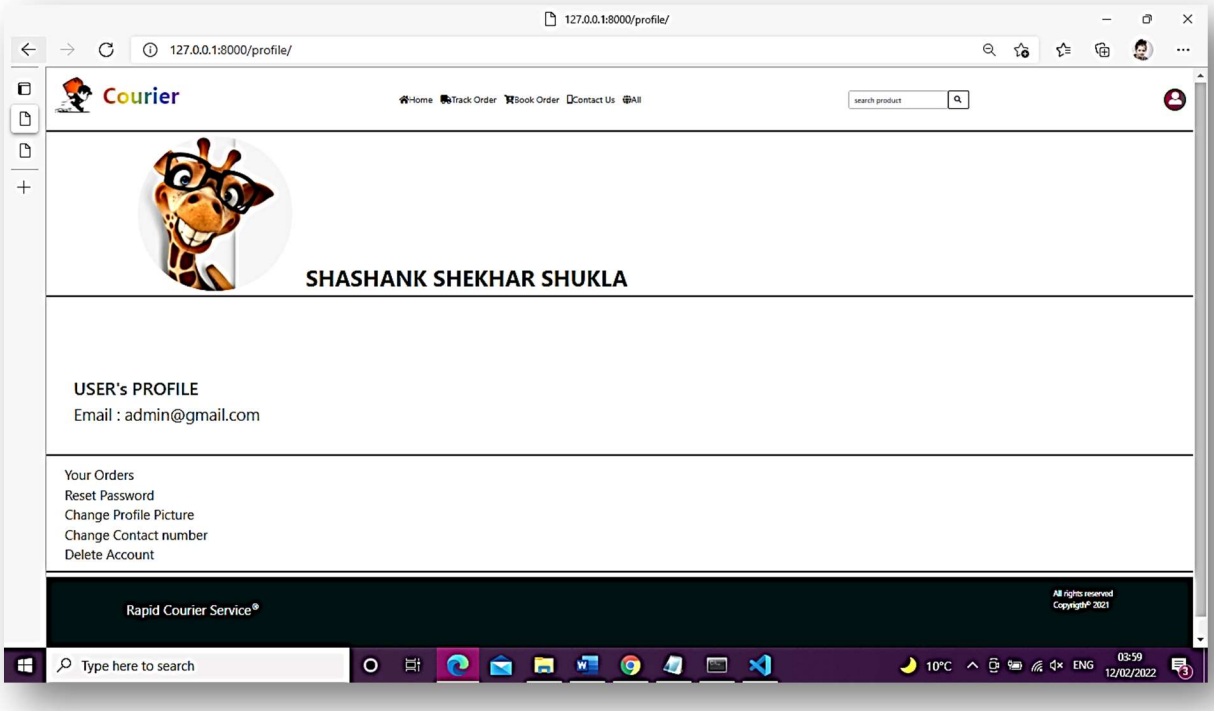


Figure 8. Displaying form data

User's Details

User's details are always available to back-end developers, it's on the Organisation to ensure the safety and security of its user's personal credentials.

Although user can't see what's happening with his credentials, and a company can't allow its users to directly access to the database, company have to ensure that his data is confidential, even to the employees of the company. Because it is found in few cases that few organisations have failed their users, those who found guilty of trading it, were punished.

Generally, a user uses same credential for every website or web application, which makes data leak or breach a potential threat to other websites that the user has access to, so a good developer will keep it in his mind and will not take the credentials from users which may potentially harm his user and other organisations, or he will make it safe against such threats.

And Django ensures the safety of the form data by accepting the data in pieces not as whole, so if anyhow it is stolen, it will be burden and nearly impossible (due to strong encryption of browsers for POST method form data) for intruder to rearrange the stolen chunks of encrypted data.

And if any how he succeeds in stooling few packets, and for an instant of lucky time lets assume he succeeded in decrypting it, it will make no sense and will be waste of his time and effort.

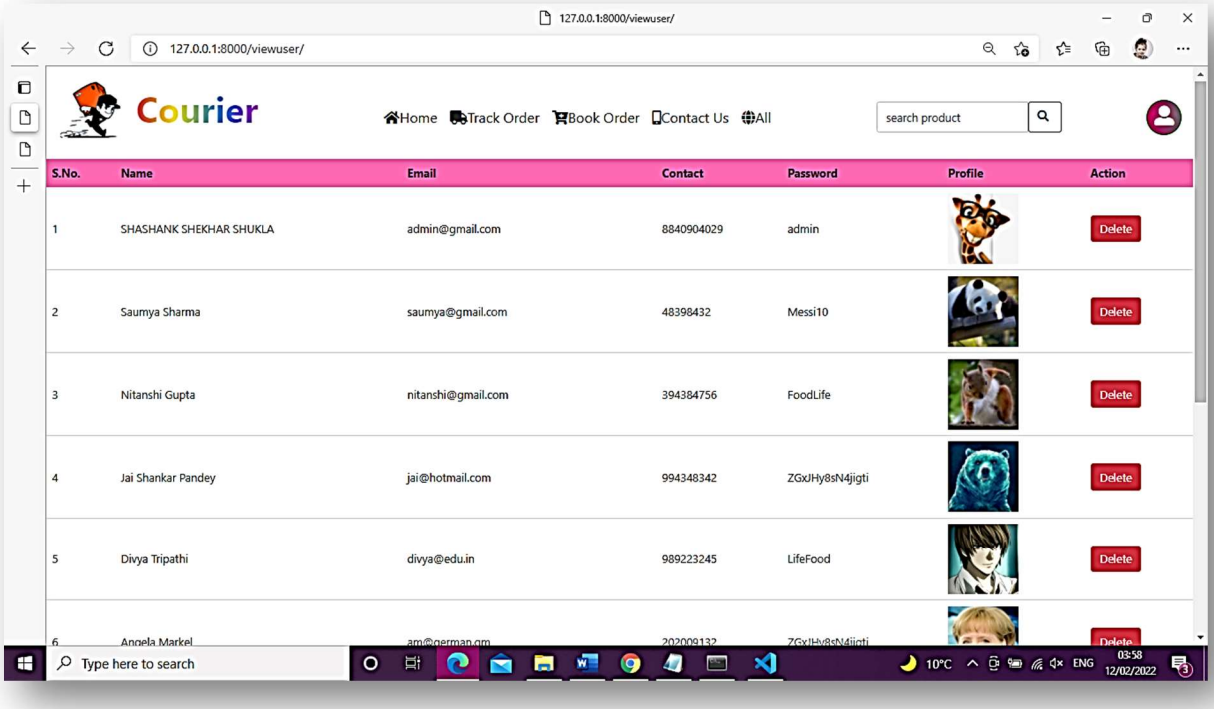
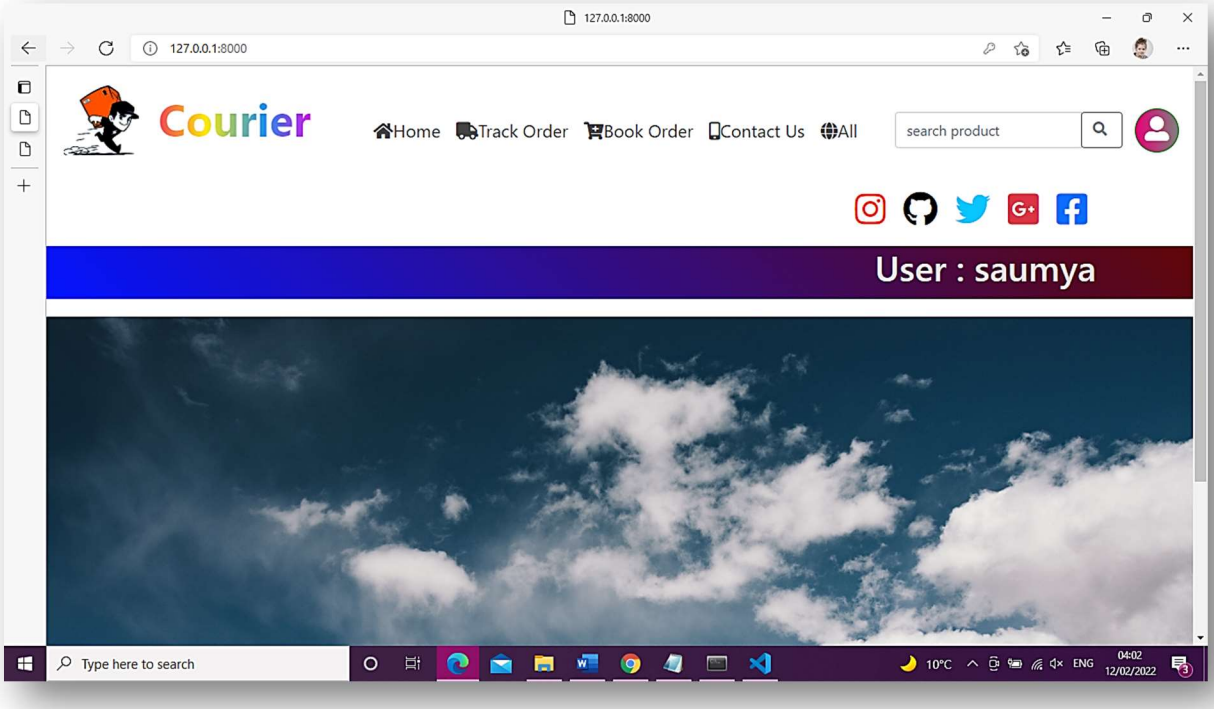


Figure 9. User details viewed by admin



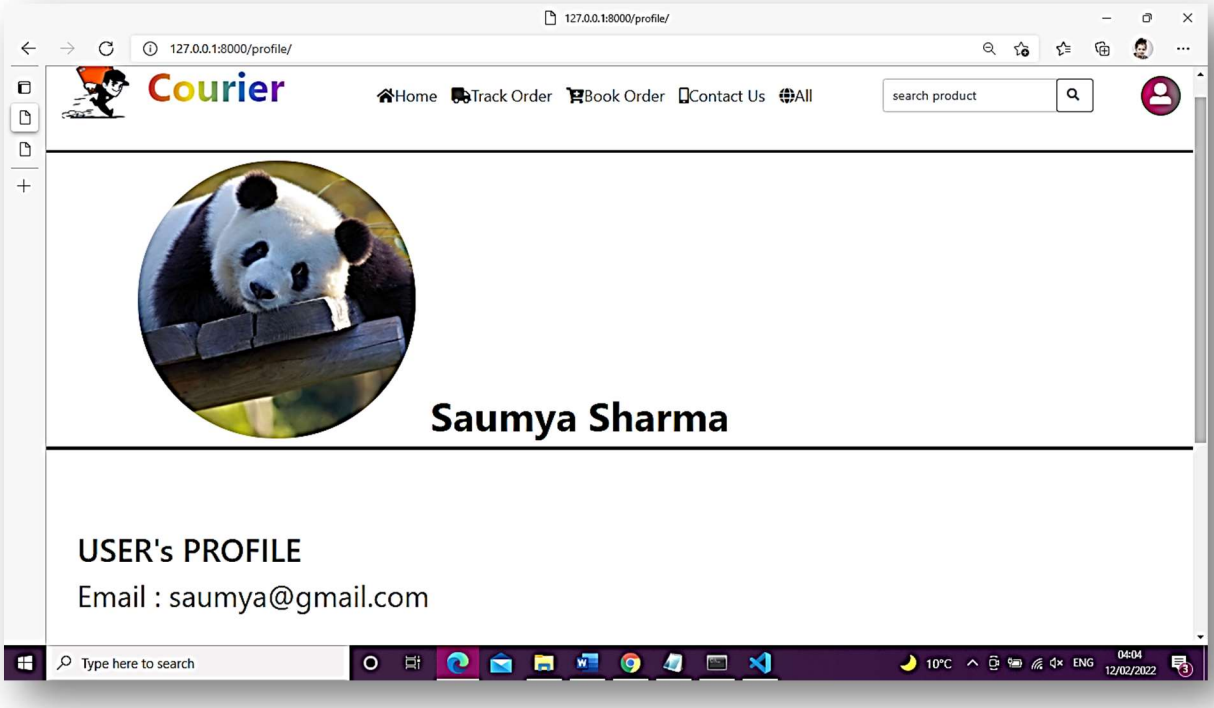


Figure 10. Partially using user's data

NOTE:

Django provide an excellent feature regarding to the safety and integrity of the user's credentials that only User can see his credentials, not even the admin, and can delete his credentials if the feature is provided by the website, admin can delete the database but not the data residing it because each data is saved in encrypted form inside the database. But it has side-effect that encryption and decryption increase response time and also the load on server. So I used it wisely.

Admin's Credential

An admin's credential is a black diamond in between the gems, as it gives anyone the superpower, here, the rights to modify the data existing in the database. So, it must be kept secret, which is obvious thing, but some developers use **"admin"** as their username and very weak password for their website.

So, Django provides secure IP based log-in feature, which makes the weak credentials very strong, so, someone can log-in as **"admin"** only if his IP address belongs to a valid range of IP addresses or have one of the specified IP addresses.

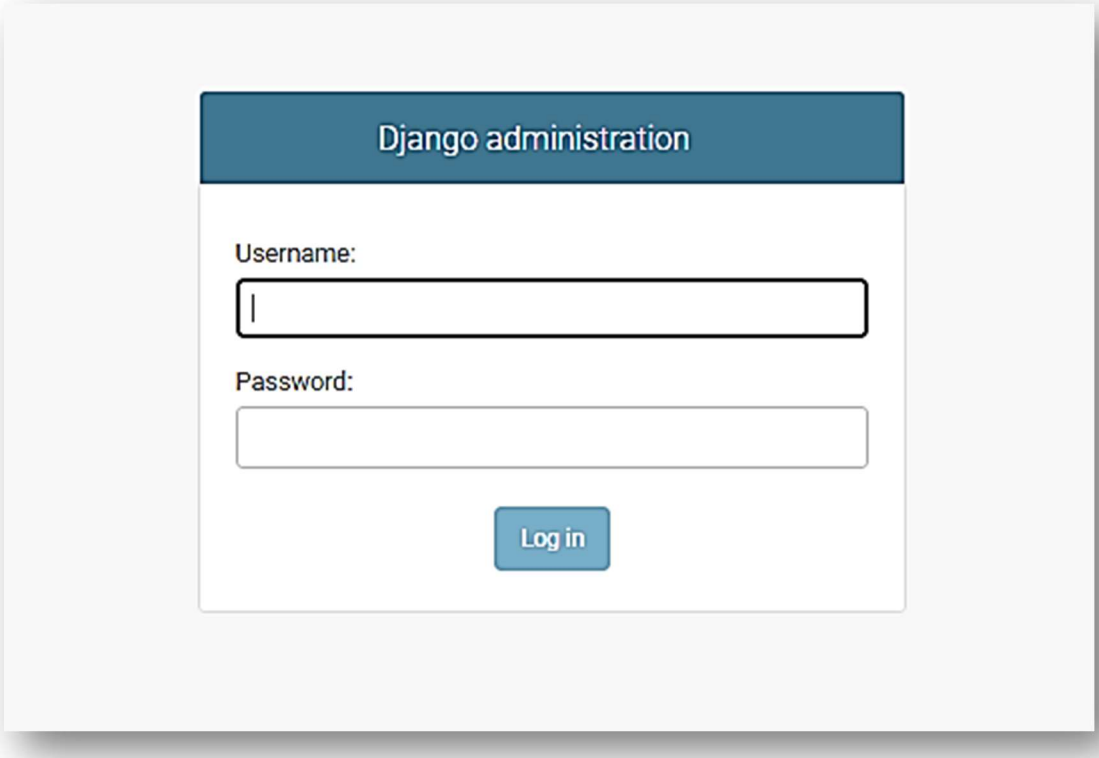


Figure 11. Secure Admin Authorization

Database:

Database are special type of software, which mainly used in dealing with website information. Primary function of it deals with long-term-storing, updating, creating and deleting data.

Database is very important. For a dynamic website to function, basically two components needed very much, which front-end and back-end of site. In the server-side programming contributes to the functioning of websites.

It plays a vital role in back-end of website. Since the data of site are loaded there and actively fetching data from it. Moreover, its inevitable topic.

Django officially supports the following Relational databases:

PostgreSQL

MariaDB

MySQL

Oracle

SQLite

There are also a number of databases backends provided by third parties.

Relational databases unofficially supported by Django:

Cockroach.

Firebird.

Google Cloud Spanner.

IBM DB2.

Microsoft SQL Server & Azure SQL.

SAP (Sybase) SQL Anywhere.

Django attempts to support as many features as possible on all database backends. However, not all database backends are alike, and we've had to make design decisions on which features to support and which assumptions we can make safely.

The Django configuration to connect to a database is done inside the setting.py file of a Django project in the DATABASES variable.

If we open the settings.py file of a Django project we'll notice the DATABASES variable has a default Python dictionary with the values

```
from pathlib import Path

BASE_DIR = Path(__file__).resolve().parent.parent


DATABASES = {
        'default': {
                'ENGINE': 'django.db.backends.sqlite3',
                'NAME': BASE_DIR / 'db.sqlite3',
        }
}
```

I used SQLite DB for my project because SQLite doesn't require additional credentials or Python packages to establish a Django database connection as it comes pre-set and supports all basic commands of SQL.

Using SQLite DB made my website fast, lite-weight, easy to configure and highly portable.

The Django DATABASES variable defines key-value pairs. Each key represents a database reference name and the value is a Python dictionary with the database

connection parameters. The default reference name is used to indicate that any database related operation declared in a Django project be executed against this connection. This means that unless otherwise specified, all database CRUD (Create-Read-Update-Delete) operations are done against the database defined with the default key.

The database connection parameters for the default database in this case are the keys ENGINE and NAME, which represent a database engine (i.e. brand like Oracle, etc.) and the name of the database instance, respectively.

The most important parameter of a Django database connection is the ENGINE value. The Django application logic associated with a database is platform neutral, which means that you always write database CRUD operations in the same way irrespective of the database selection. Nevertheless, there are minor differences between CRUD operations done against different databases which need to be taken into account.

Django takes care of this issue by supporting different backends or engines. Therefore, depending on the database brand you plan to use for a Django application, the ENGINE value has to be one of the values illustrated in table below.

Database		Django ENGINE value	Required Django package
Officially supported	MariaDB	†django.db.backends.mysql	None, it's included with Django
	MySQL	django.db.backends.mysql	None, it's included with Django
	Oracle	django.db.backends.oracle	None, it's included with Django
	PostgreSQL	django.db.backends.postgresql_psycopg2	None, it's included with Django
	SQLite	django.db.backends.sqlite3	None, it's included with Django

Unofficially supported	Cockroach	django_cockroachdb	django-cockroachdb
	Firebird	django.db.backends.firebird	django-firebird
	Google Cloud Spanner	django_spanner	django-google-spanner
	IBM DB2	ibm_db_django	ibm_db_django
	Microsoft SQL Server & Azure SQL	mssql	mssql-django
	SAP (Sybase) SQL Anywhere	sqlany_django	sqlany_django

Table 1. Django ENGINE value for different databases

† MariaDB is supported through the same built-in ENGINE used by MySQL: django.db.backends.mysql.

The Django database connection parameter NAME is used to identify a database instance, and its value convention can vary depending on the database brand. For example, in above code for the SQLite database the NAME value indicates the location of a flat file, whereas for a MySQL database it indicates the logical name of a database instance.

In addition to the ENGINE and NAME connection parameters, many other database parameters can be used to influence how Django connects to a database.

NOTE:

Database brand specific connection parameters (e.g., PostgreSQL service name) can be configured in Django through the OPTIONS parameter dictionary. (e.g., "OPTIONS": {“service”: ...))

PostgreSQL connection settings

To connect using a service name from the connection service file and a password from the password file, you must specify them in the OPTIONS part of your database configuration in DATABASES:

In settings.py

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql',
        'OPTIONS': {
            'service': 'my_service',
            'passfile': '.my_pgpass',
        },
    }
}
```

In .pg_service.conf

```
[my_service]
host=localhost
user=USER
dbname=NAME
port=5432
```

in .my_pgpass

```
localhost:5432:NAME:USER:PASSWORD
```

Creating your database

You can create your database using the command-line tools and this SQL:

```
CREATE DATABASE <dbname> CHARACTER SET utf8;
```

This ensures all tables and columns will use UTF-8 by default.

Here's a sample configuration which uses a MySQL option file:

```
# settings.py
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'OPTIONS': {
            'read_default_file': '/path/to/my.cnf',
        },
    }
}

# my.cnf
[client]
database = NAME
user = USER
password = PASSWORD
default-character-set = utf8
```

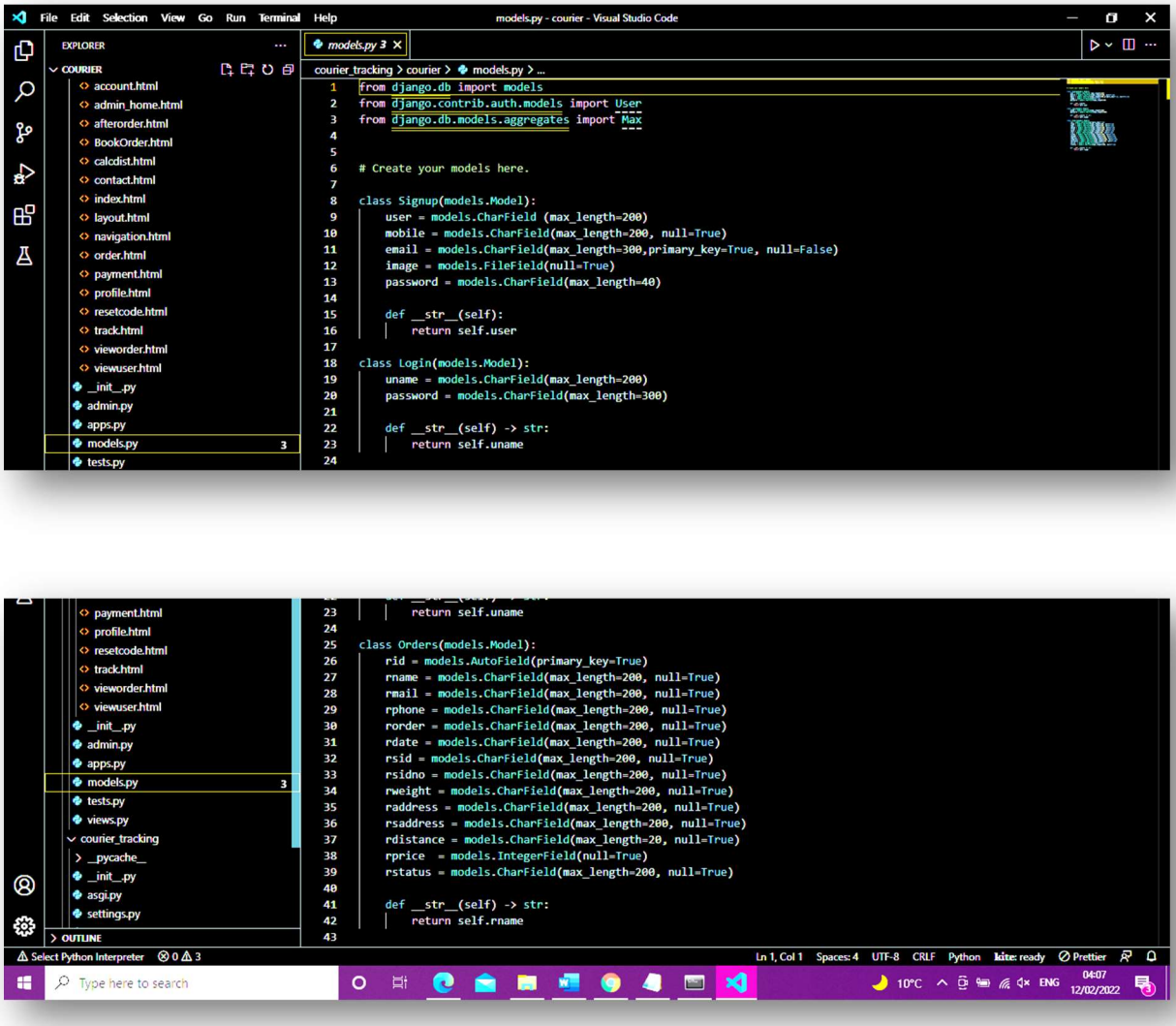


Figure 12. Creating tables in database

```
21
22  def sign(request):
23      global user, Admin, em
24      if request.method == 'POST' :
25          u_name = request.POST['uname']
26          p_code = request.POST['passcode']
27          obj = Login.objects.filter(uname=u_name, password=p_code).first()
28          if obj:
29              user = str(u_name)
30              em = str(u_name)
31              user = user[:user.index('@')]
32              if user == "admin":
33                  Admin=True
34              return HttpResponseRedirect(reverse('index'))
35          else:
36              return render(request, 'account.html',{"user" : user, "Admin":Admin })
37      else:
38          return render(request, 'account.html',{"user" : user, "Admin":Admin } )
```

Figure 13. Querying database

Security:

Cross site scripting (XSS) protection

XSS attacks allow a user to inject client side scripts into the browsers of other users. This is usually achieved by storing the malicious scripts in the database where it will be retrieved and displayed to other users, or by getting users to click a link which will cause the attacker's JavaScript to be executed by the user's browser. However, XSS attacks can originate from any untrusted source of data, such as cookies or web services, whenever the data is not sufficiently sanitized before including in a page.

Using Django templates protects you against the majority of XSS attacks. However, it is important to understand what protections it provides and its limitations.

Django templates escape specific characters which are particularly dangerous to HTML. While this protects users from most malicious input, it is not entirely foolproof. For example, it will not protect the following:

<style class={{ var }}>...</style>

If var is set to 'class1 onmouseover=javascript:func()', this can result in unauthorized JavaScript execution, depending on how the browser renders imperfect HTML. (Quoting the attribute value would fix this case.)

It is also important to be particularly careful when using is_safe with custom template tags, the safe template tag, mark_safe, and when autoescape is turned off.

In addition, if you are using the template system to output something other than HTML, there may be entirely separate characters and words which require escaping.

You should also be very careful when storing HTML in the database, especially when that HTML is retrieved and displayed.

Cross site request forgery (CSRF) protection

CSRF attacks allow a malicious user to execute actions using the credentials of another user without that user's knowledge or consent.

Django has built-in protection against most types of CSRF attacks, providing you have enabled and used it where appropriate. However, as with any mitigation technique, there are limitations. For example, it is possible to disable the CSRF module globally or for particular views. You should only do this if you know what you are doing. There are other limitations if your site has subdomains that are outside of your control.

CSRF protection works by checking for a secret in each POST request. This ensures that a malicious user cannot "replay" a form POST to your website and have another logged in user unwittingly submit that form. The malicious user would have to know the secret, which is user specific (using a cookie).

When deployed with HTTPS, CsrfViewMiddleware will check that the HTTP referer header is set to a URL on the same origin (including subdomain and port). Because HTTPS provides additional security, it is imperative to ensure connections use HTTPS where it is available by forwarding insecure connection requests and using HSTS for supported browsers.

Be very careful with marking views with the csrf_exempt decorator unless it is absolutely necessary.

SQL injection protection

SQL injection is a type of attack where a malicious user is able to execute arbitrary SQL code on a database. This can result in records being deleted or data leakage.

Django's querysets are protected from SQL injection since their queries are constructed using query parameterization. A query's SQL code is defined separately from the query's parameters. Since parameters may be user-provided and therefore unsafe, they are escaped by the underlying database driver.

Django also gives developers power to write raw queries or execute custom sql. These capabilities should be used sparingly and you should always be careful to properly escape any parameters that the user can control. In addition, you should exercise caution when using extra() and RawSQL.

Clickjacking protection

Clickjacking is a type of attack where a malicious site wraps another site in a frame. This attack can result in an unsuspecting user being tricked into performing unintended actions on the target site.

Django contains clickjacking protection in the form of the X-Frame-Options middleware which in a supporting browser can prevent a site from being rendered inside a frame. It is possible to disable the protection on a per view basis or to configure the exact header value sent.

The middleware is strongly recommended for any site that does not need to have its pages wrapped in a frame by third party sites, or only needs to allow that for a small section of the site.

SSL/HTTPS

It is always better for security to deploy your site behind HTTPS. Without this, it is possible for malicious network users to sniff authentication credentials or any other information transferred between client and server, and in some cases – active network attackers – to alter data that is sent in either direction.

If you want the protection that HTTPS provides, and have enabled it on your server, there are some additional steps you may need:

If necessary, set SECURE_PROXY_SSL_HEADER, ensuring that you have understood the warnings there thoroughly. Failure to do this can result in CSRF vulnerabilities, and failure to do it correctly can also be dangerous!

Set SECURE_SSL_REDIRECT to True, so that requests over HTTP are redirected to HTTPS.

Use ‘secure’ cookies.

If a browser connects initially via HTTP, which is the default for most browsers, it is possible for existing cookies to be leaked. For this reason, you should set your SESSION_COOKIE_SECURE and CSRF_COOKIE_SECURE settings to True. This instructs the browser to only send these cookies over HTTPS connections. Note that this will mean that sessions will not work over HTTP, and the CSRF protection will prevent any POST data being accepted over HTTP (which will be fine if you are redirecting all HTTP traffic to HTTPS).

Use HTTP Strict Transport Security (HSTS)

HSTS is an HTTP header that informs a browser that all future connections to a particular site should always use HTTPS. Combined with redirecting requests over HTTP to HTTPS, this will ensure that connections always enjoy the added security of SSL provided one successful connection has occurred. HSTS may either be configured with SECURE_HSTS_SECONDS, SECURE_HSTS_INCLUDE_SUBDOMAINS, and SECURE_HSTS_PRELOAD, or on the web server.

Host header validation

Django uses the Host header provided by the client to construct URLs in certain cases. While these values are sanitized to prevent Cross Site Scripting attacks, a fake Host value can be used for Cross-Site Request Forgery, cache poisoning attacks, and poisoning links in emails.

Because even seemingly-secure web server configurations are susceptible to fake Host headers, Django validates Host headers against the ALLOWED_HOSTS setting in the django.http.HttpRequest.get_host() method.

This validation only applies via get_host(); if your code accesses the Host header directly from request.META you are bypassing this security protection.

Session security

Similar to the CSRF limitations requiring a site to be deployed such that untrusted users don't have access to any subdomains, django.contrib.sessions also has limitations. See the session topic guide section on security for details.

5 METHODOLOGY

For this project, most of the requirements were unknown to me, and I had no prior experience in web-development, So I found the Agile methodology best suitable to use for the development of this project as this methodology allows to start development even if the requirement is not well known, and it is best for modular designing of the Software, so, being alone, I found it best suitable for me. There are several other advantages of agile methodology.

Agile is a structured and iterative approach to project management and product development. On this system of approaches, flexible project management methodologies (Scrum, Kanban, XP, and others) are built.

Agile methodologies are an alternative to Waterfall or traditional sequential development.

In short, Agile is a time-focused philosophy that allows creating a project incrementally, dividing it into small pieces. One of its main benefits is the ability to adapt and change at any step and to supply only relevant products to the market.

There are no exact stages; time is time-boxed into sprints. A sprint is a time allocated for particular tasks and defined deliverables. The tasks' value is supposed to be defined by a customer, who's deeply involved in the development process. A sprint usually is measured in weeks.

The Agile approach is all about customer's presence and controlling (not necessarily physical). The customer should be ready to dedicate some time to reviewing sprint outcomes, assessment and (re)prioritizing. A client can test the basic product version before a final release or even put a basic version on the market. That's a really great approach for the markets where being first means everything. Also, a client can change the project requirements on and off.

Some of the Agile principle:

. People and cooperation are more important than processes and tools

. The working product is more important than documentation

. Collaboration with a customer is more important than negotiating the terms of the contract

. Readiness for change is more important than following the original plan

. An agile company is very flexible quickly adapts to changes, iterates less while implementing faster, and is able to seize new opportunities as they appear

. High level of interaction between project team members

As being this much beneficial, it has some disadvantages too:

. The very high degree of customer involvement, while great for the project, may present problems for some customers who simply may not have the time or interest for this type of participation

. Risk of endless product changes

6 IMPLEMENTATION

Create the project

```
$ django-admin startproject courier
```

1. Use the django-admin tool to generate a project folder, the basic file templates, and manage.py, which serves as your project management script.
2. Use manage.py to create one or more applications.

Note: A website may consist of one or more sections. For example, main site, blog, wiki, downloads area, etc. Django encourages you to develop these components as separate applications, which could then be re-used in different projects if desired.

3. Register the new applications to include them in the project.
4. Hook up the url/path mapper for each application.
5. Code all the necessary files using the knowledge of Python, HTML, CSS, JS and Django.

The development server

Now it's time to verify Django project works. Change into the outer courier directory, if haven't already, and run the following commands:

```
$ python manage.py runserver
```

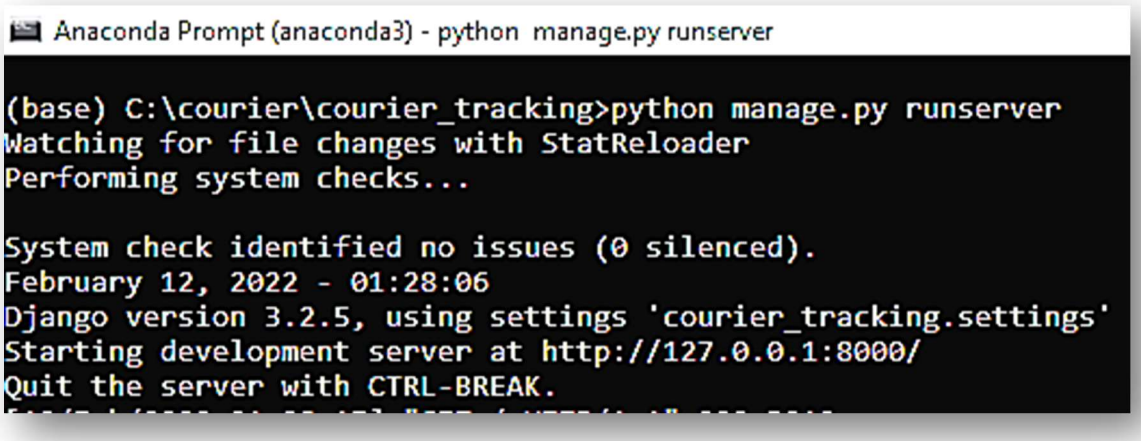


Figure 14. Deploying the website

Changing the port

By default, the runserver command starts the development server on the internal IP at port 8000.

If you want to change the server's port, pass it as a command-line argument. For instance, this command starts the server on port 8080.

```
$ python manage.py runserver 8080
```

If you want to change the server's IP, pass it along with the port. For example, to listen on all available public IPs (which is useful if you are running Vagrant or want to show off your work on other computers on the network), use:

```
$ python manage.py runserver 0:8080
```

** **:** 0 is a shortcut for 0.0.0.0.

NOTE: The development server automatically reloads Python code for each request as needed. You don't need to restart the server for code changes to take effect. However, some actions like adding files don't trigger a restart, so you'll have to restart the server in these cases.

7 TESTING

Software testing is the process of finding errors in the developed product. It also checks whether the real outcomes can match expected results, as well as aids in the identification of defects, missing requirements, or gaps.

Testing is the penultimate step before the launch of the product to the market. It includes examination, analysis, observation, and evaluation of different aspects of a product.

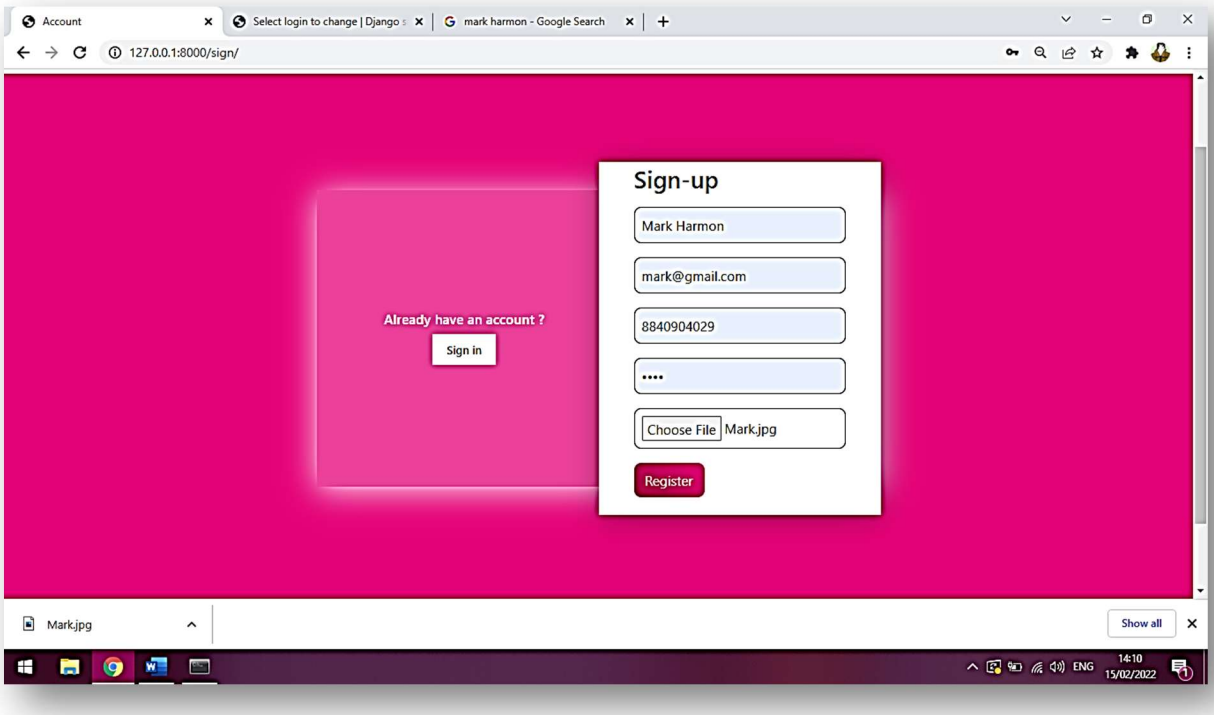
Professional software testers use a combination of manual testing with automated tools. But as it was a small project so I performed testing manually.

I PERFORMED:

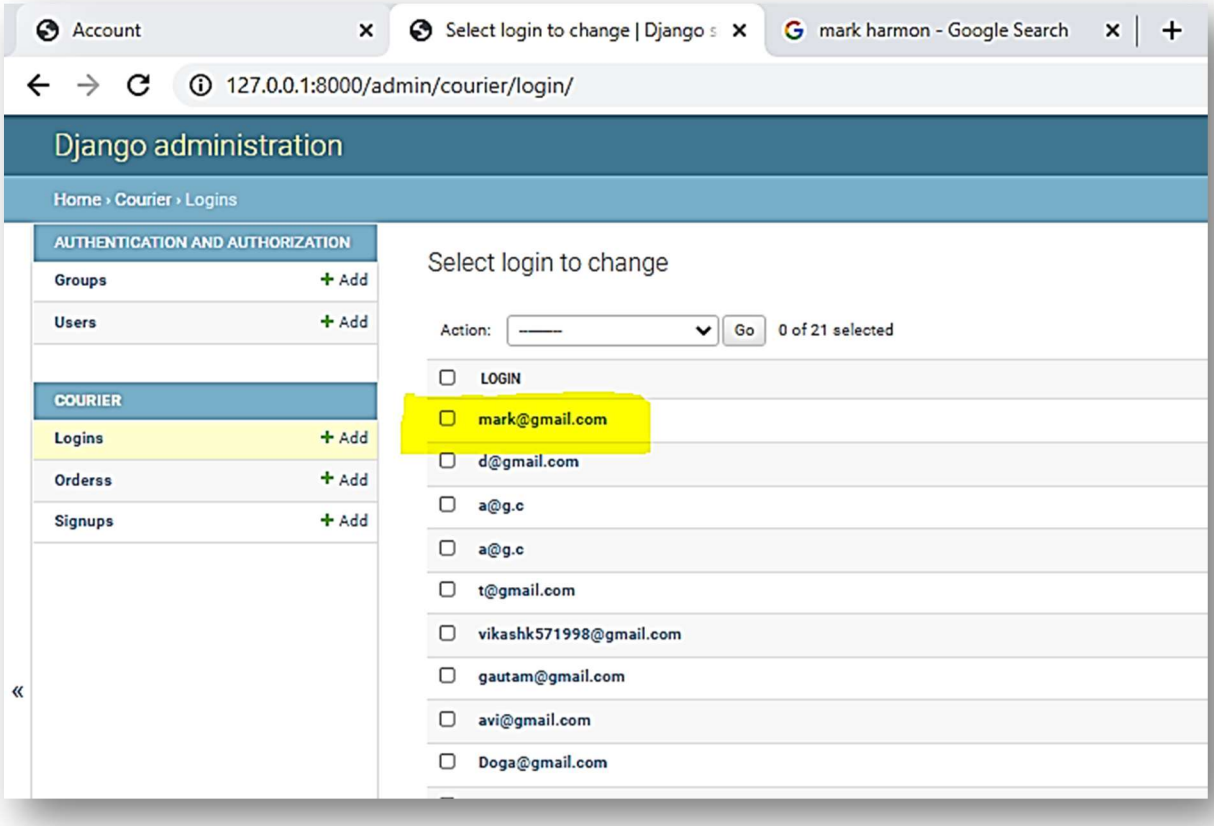
Functionality Testing

It is a process that includes several testing parameters like user interface, APIs, database testing, security testing, client and server testing and basic website functionalities. Functional testing is very convenient and it allows users to perform both manual and automated testing. It is performed to test the functionalities of each feature on the website.

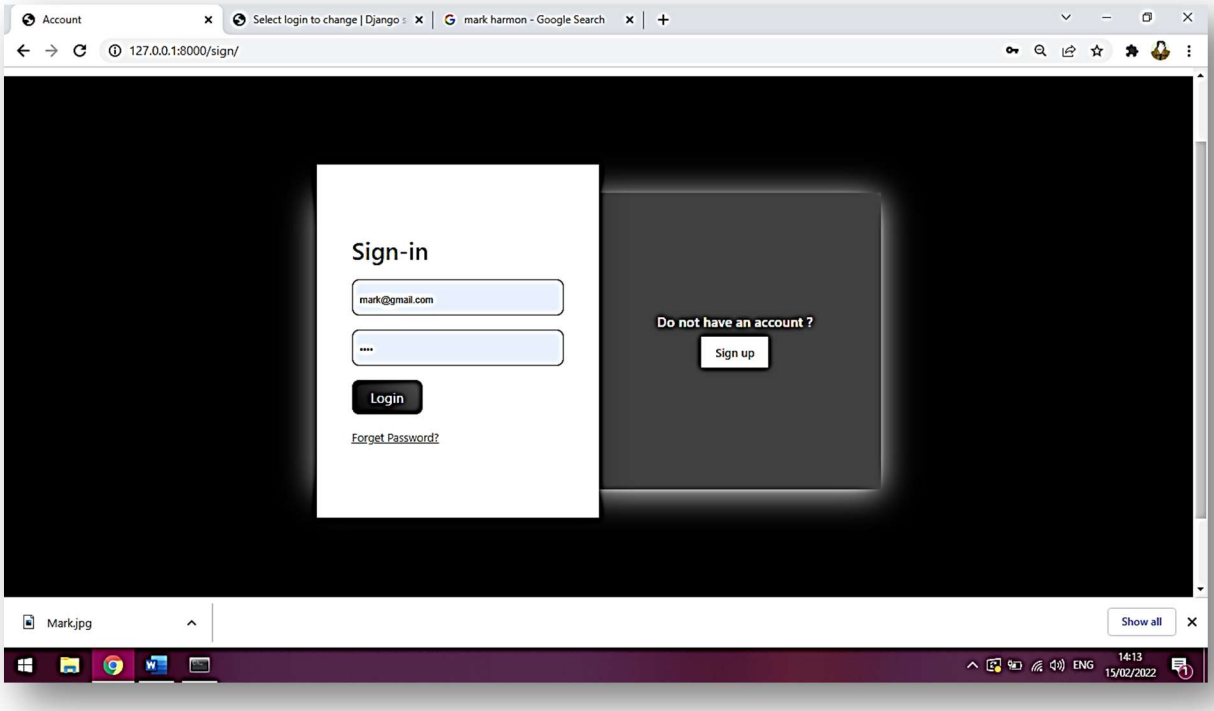
TESTING* SIGN *FUNCTIONALITIES:



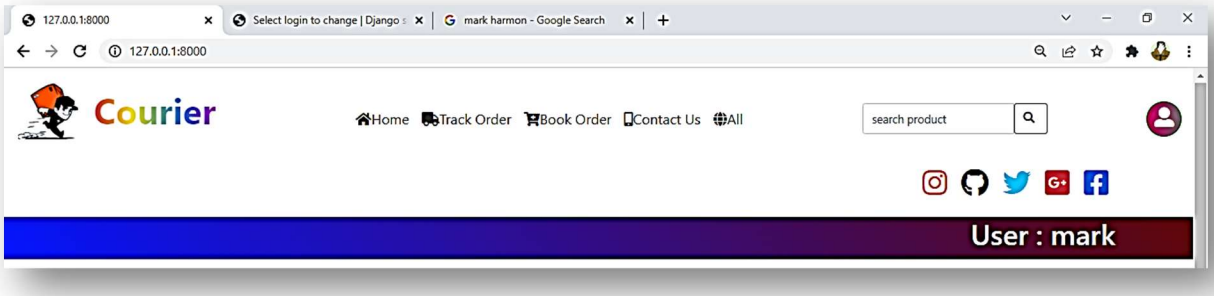
SIGN-UP PROCESS SUCCESSFUL AT FRONT-END

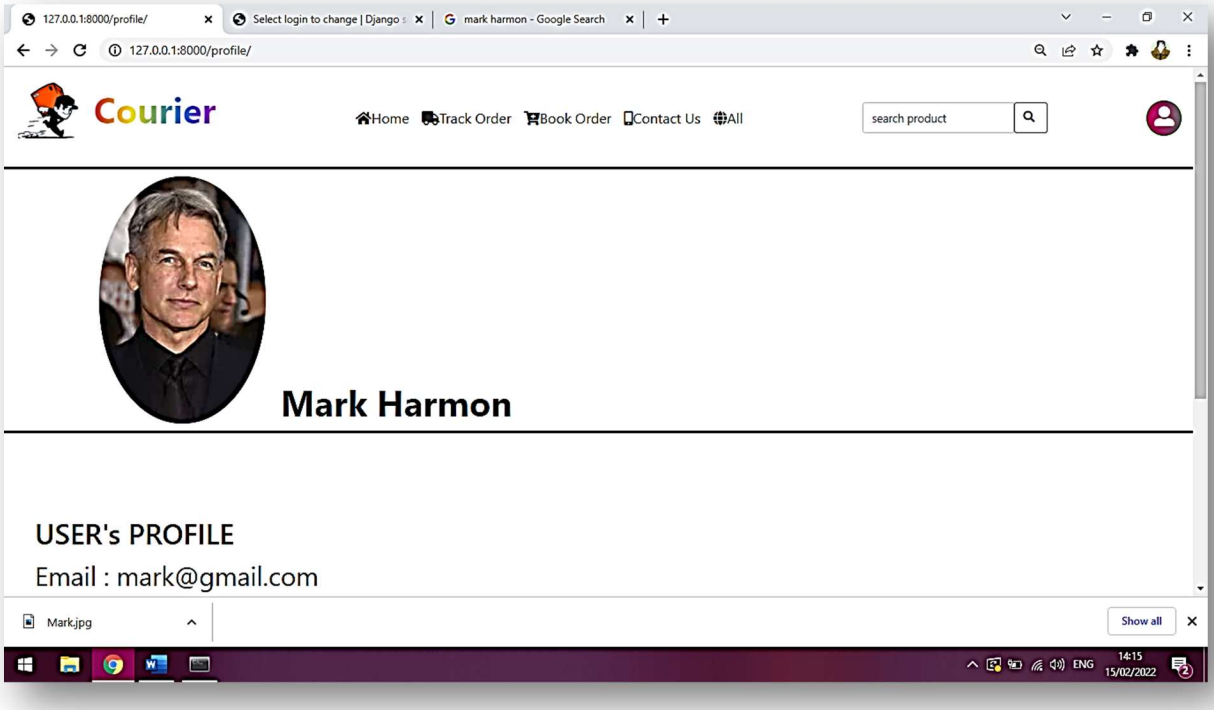


CREDENTIALS GOT SAVED AT BACK-END

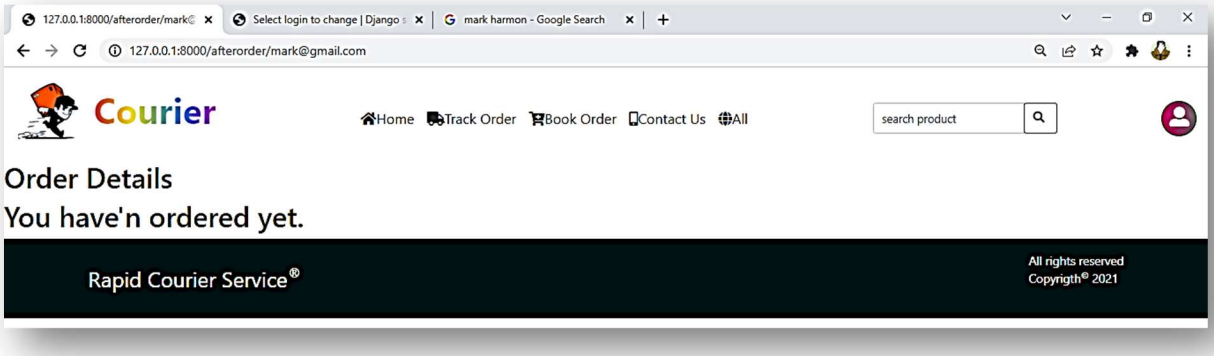


SUCCESSFULLY LOGED IN USING NEWLY CREATED ACCOUNT

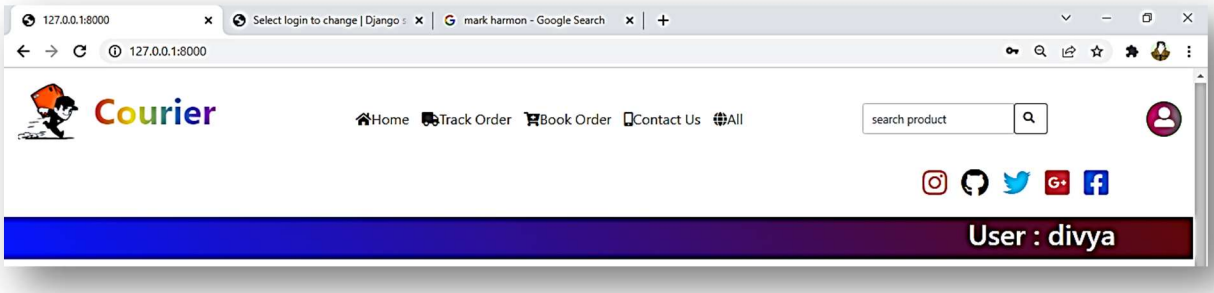


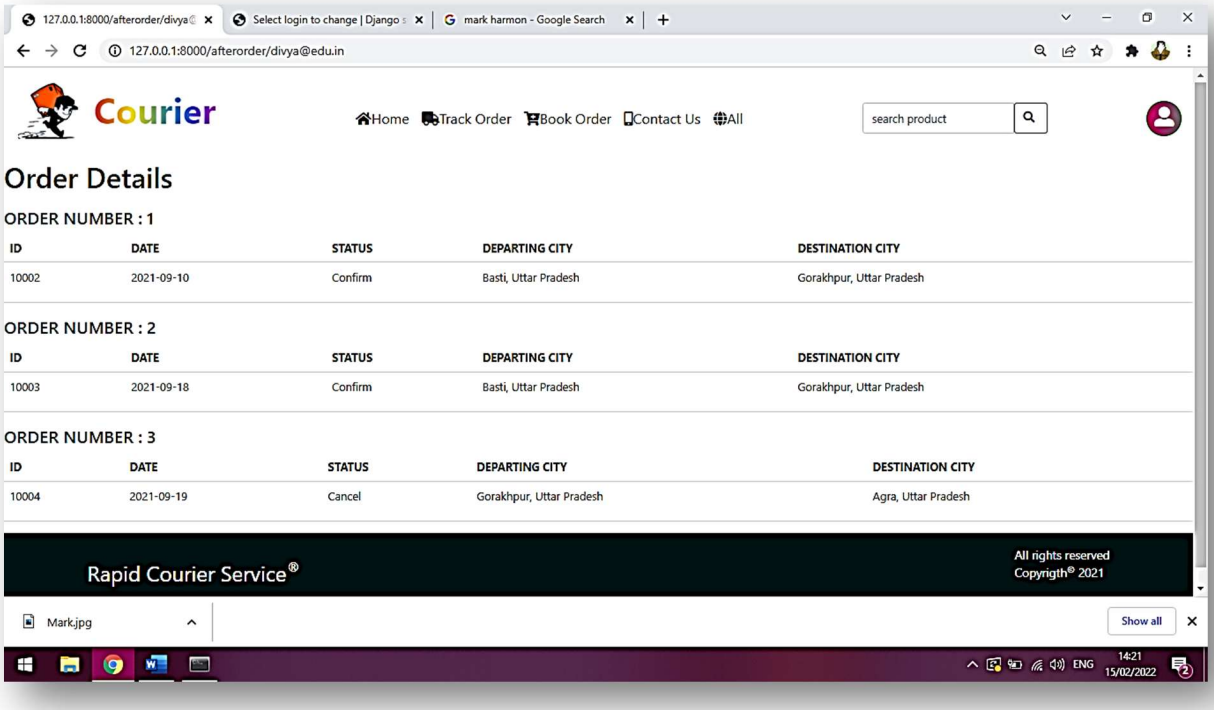


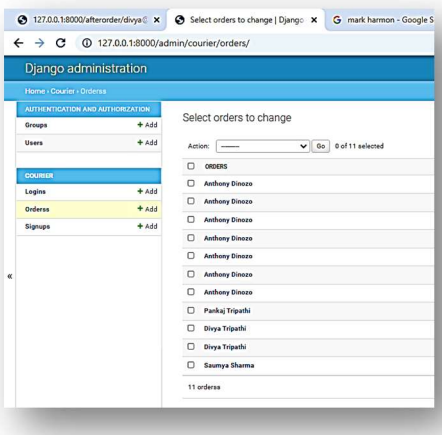
DATA IS SHOWN PROPERLY, NEW ACCOUNT HAVE NO ORDERS



ANOTHER ACCOUNT WITH ORDERS

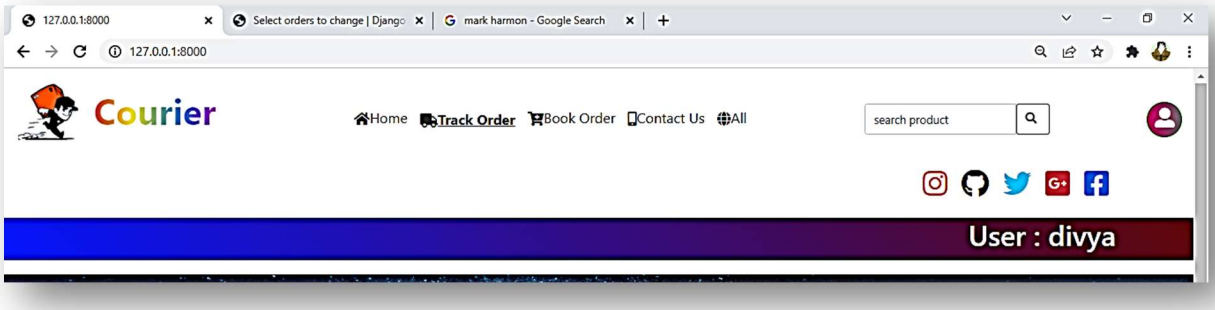




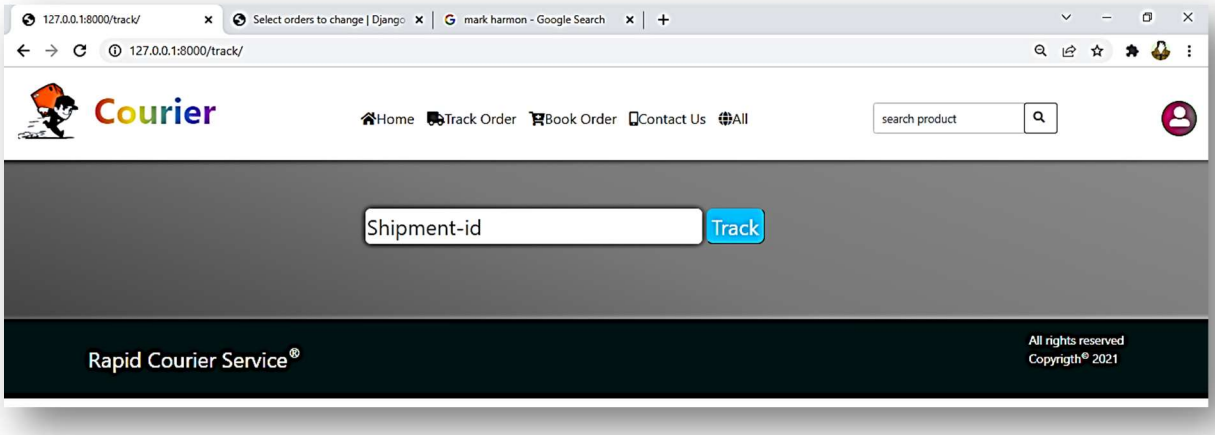


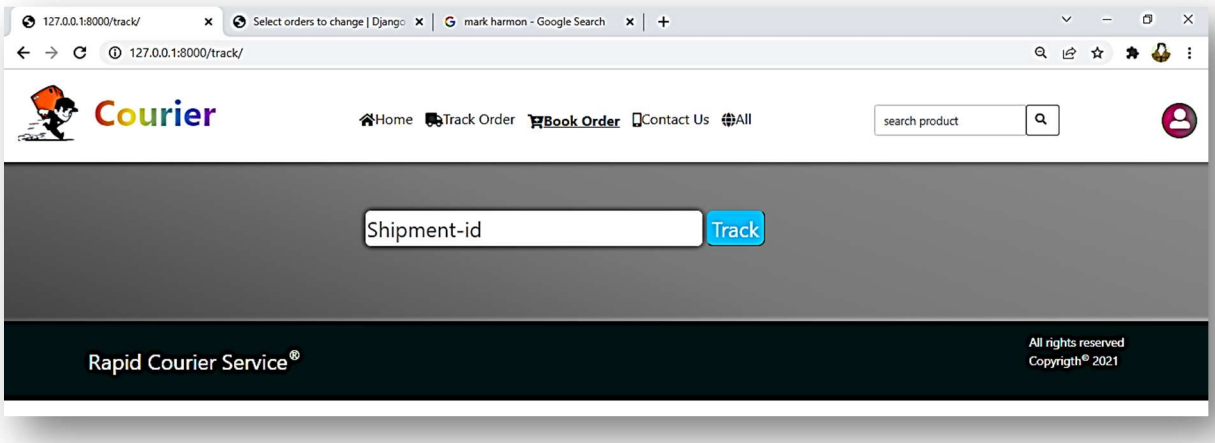
Usability Testing

It checks that Menus, buttons or Links to different pages on your site are easily visible and consistent on all webpages or not.

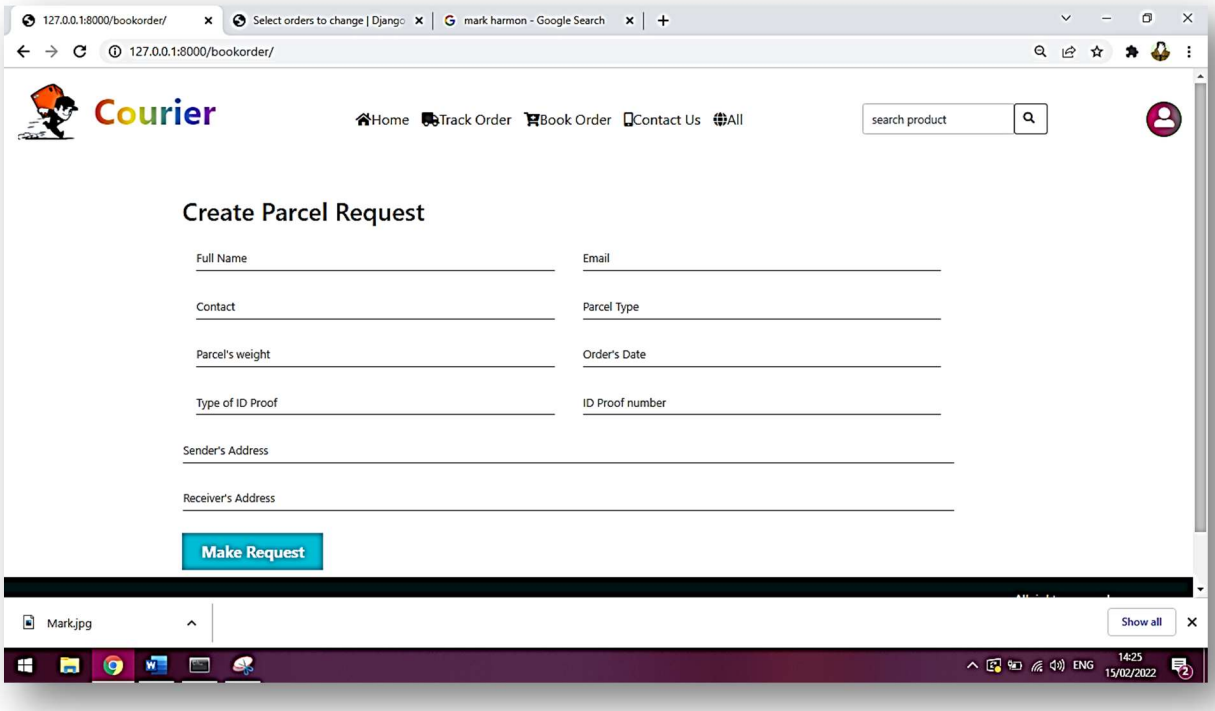


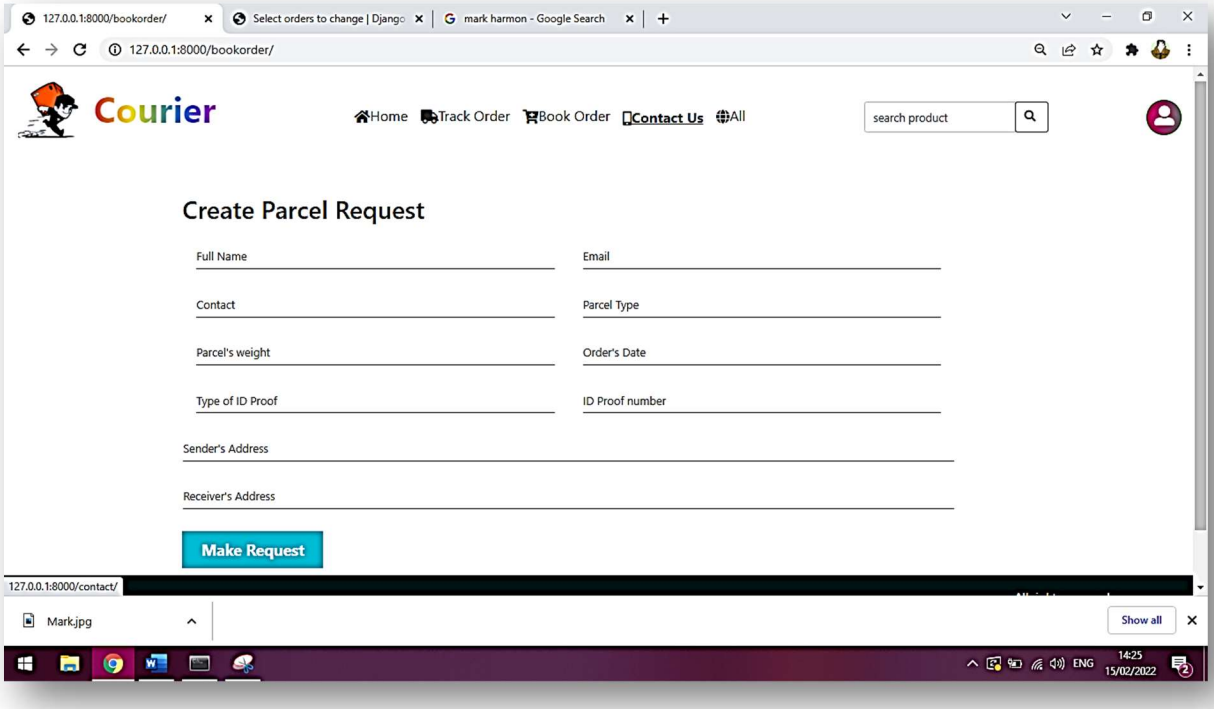
TRACK ORDER BUTTON IS WORKING FINE



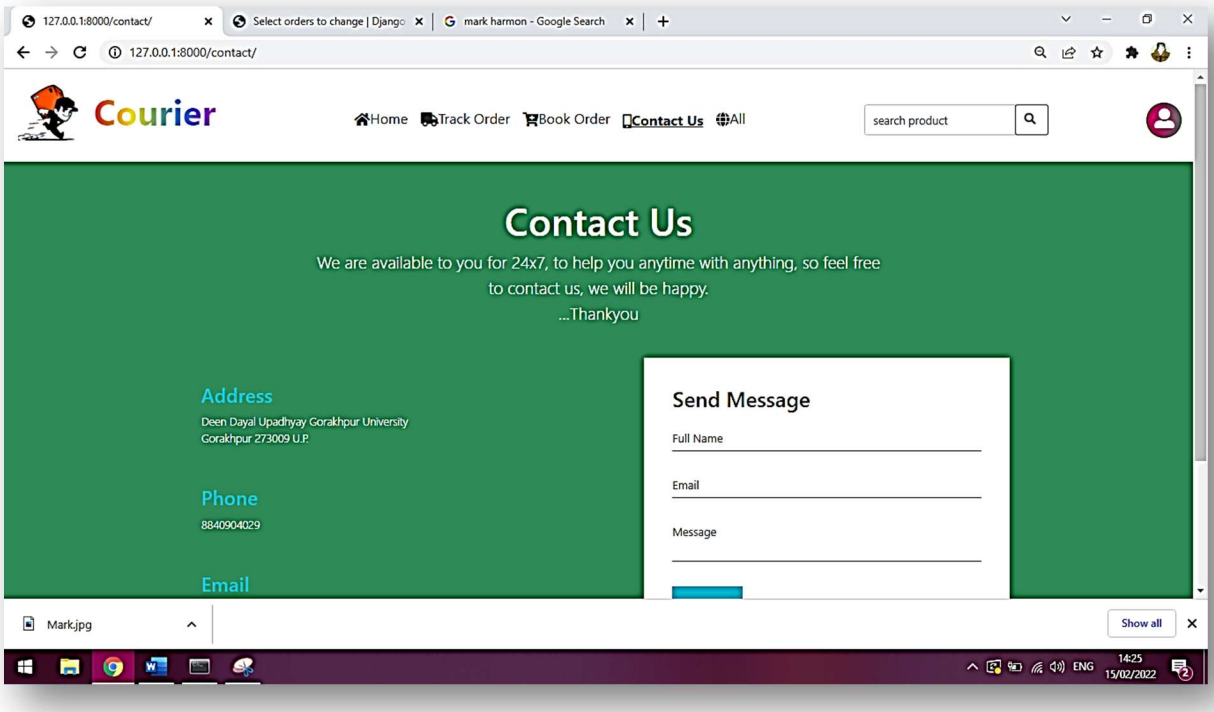


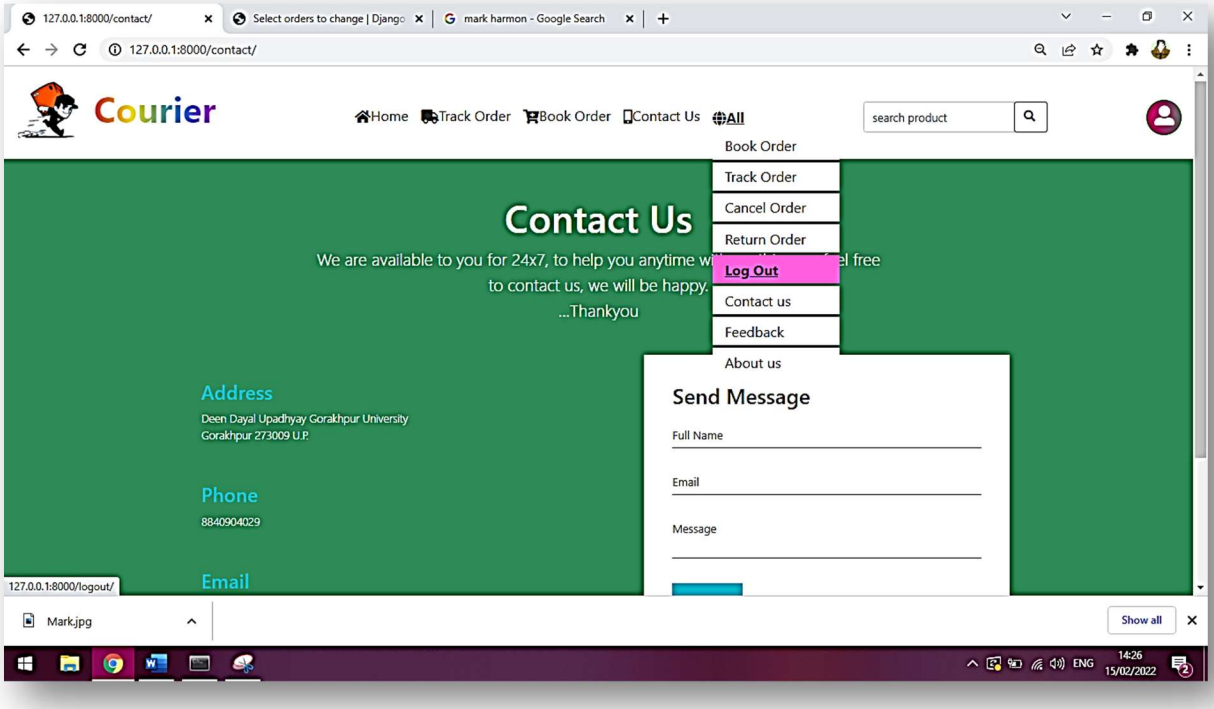
BOOK ORDER BUTTON IS WORKING FINE



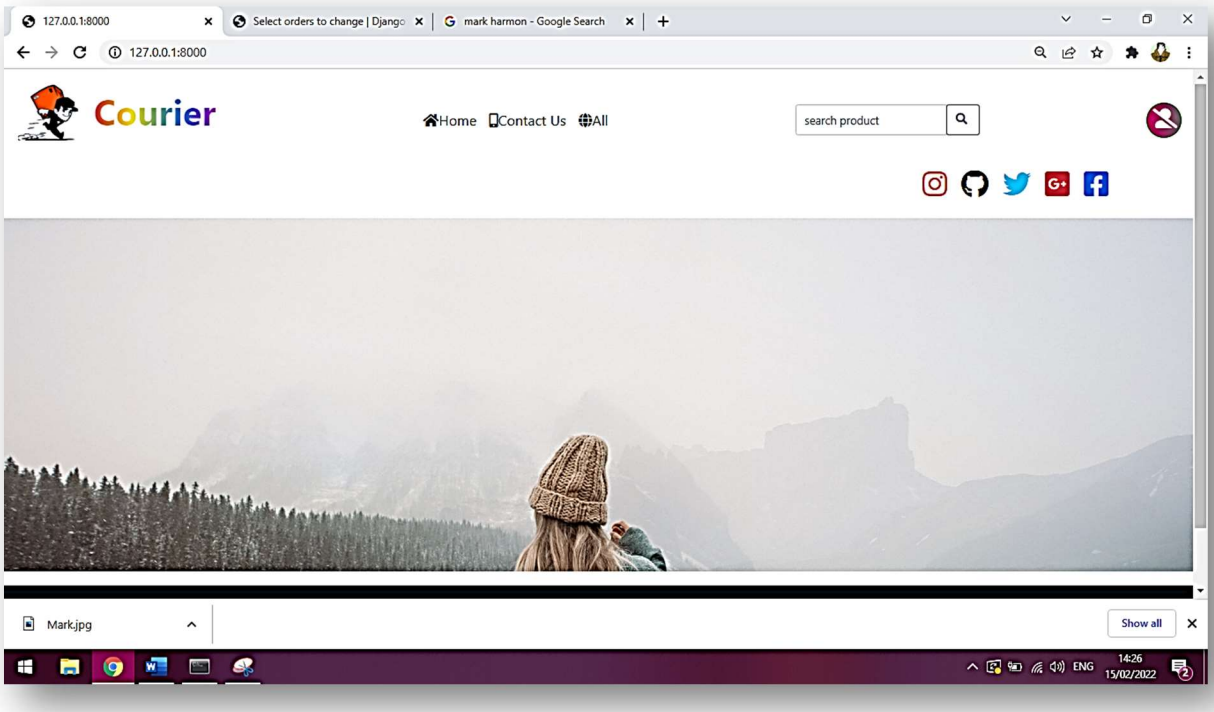


CONTACT US BUTTON IS WORKING FINE





LOG-OUT LOGGED OUT THE USER



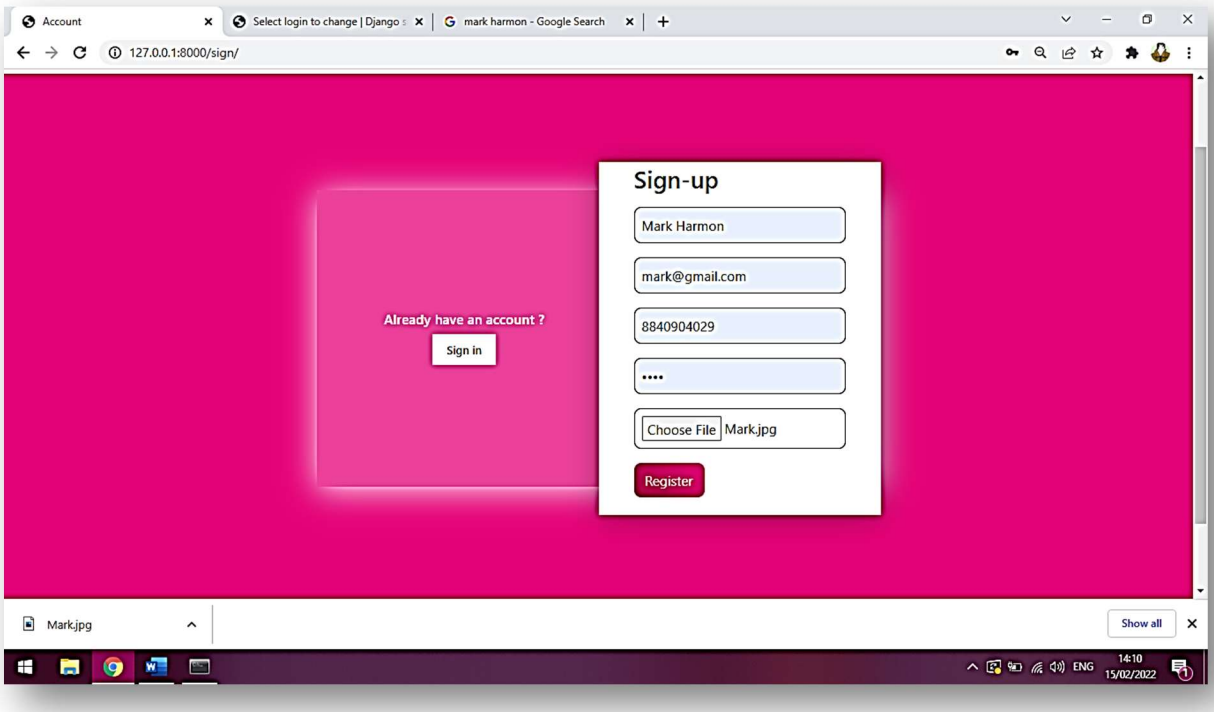
Interface Testing

It tests three areas– Application, Web and Database Server

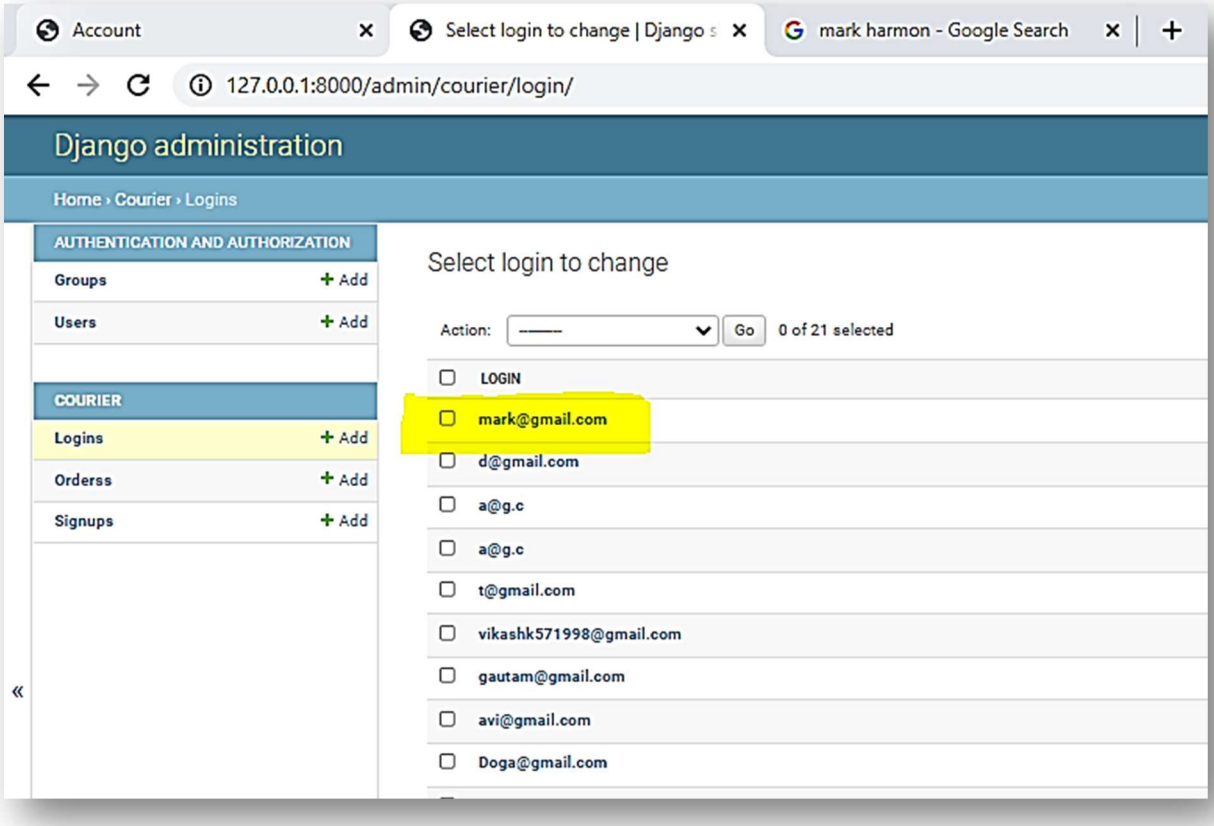
Application: Test requests are sent correctly to the Database and output at the client side is displayed correctly. Errors if any must be caught by the application and must be only shown to the administrator and not the end user.

Web Server: Test Web server is handling all application requests without any service denial.

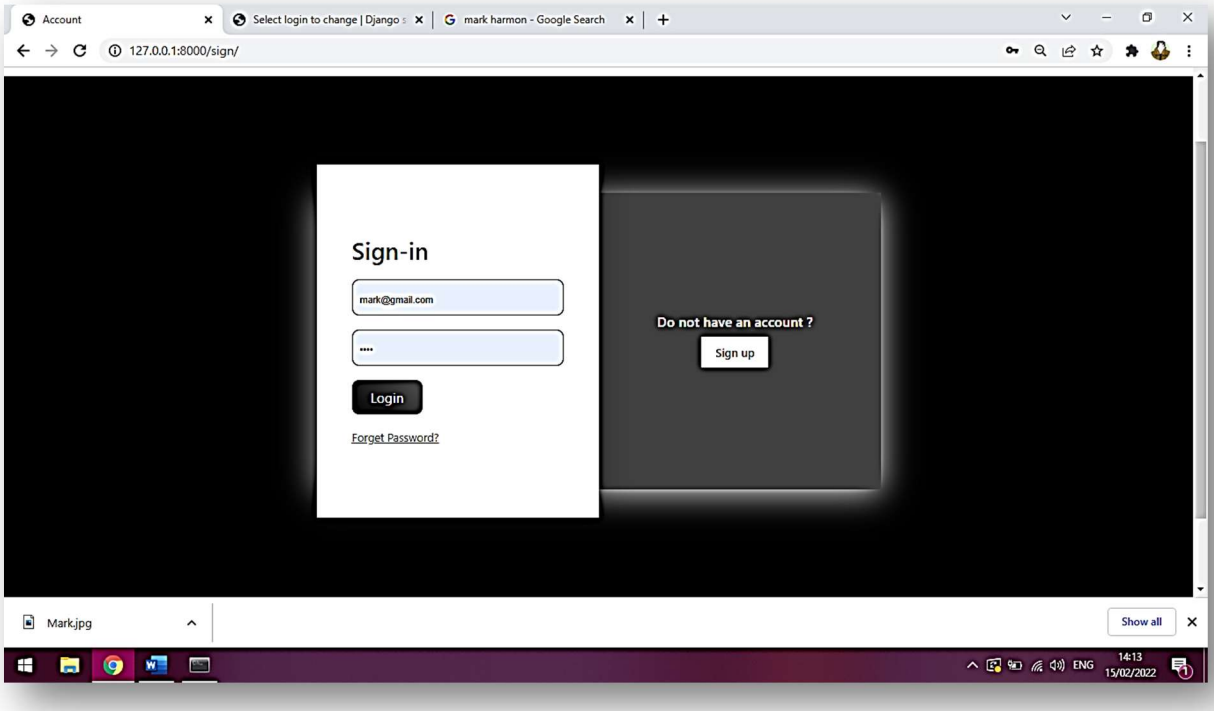
Database Server: Make sure queries sent to the database give expected results.

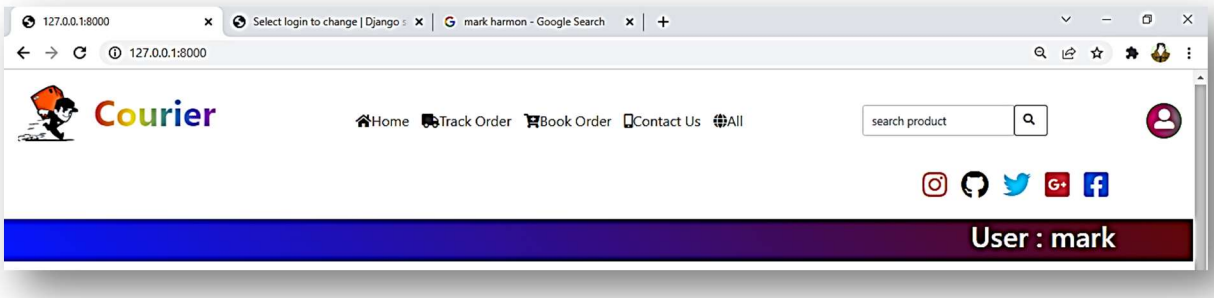


TESTING THE FRONT-END INTERFACE TO CREATE NEW ACCOUNT

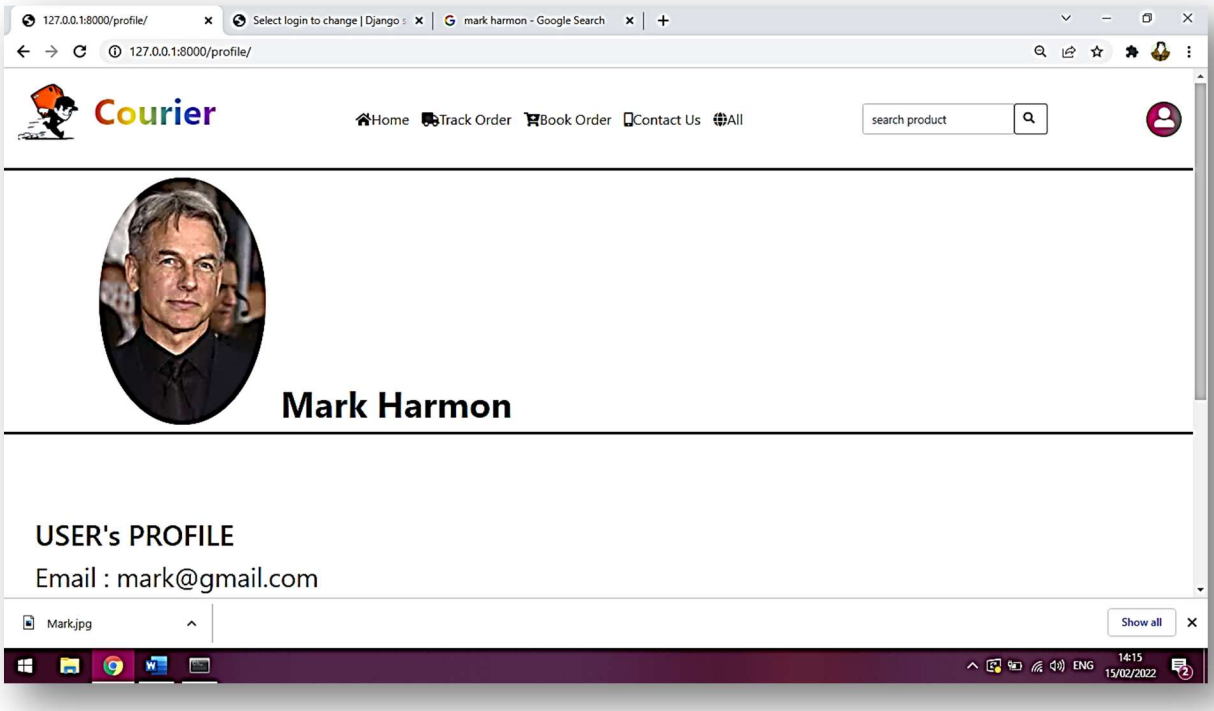


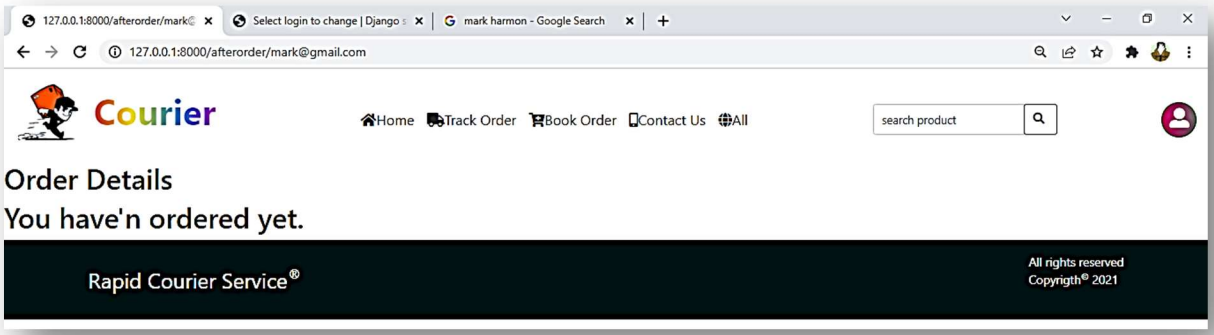
NEW CREDENTIAL INSERTED SUCCESSFULLY





INTERFACE IS WORKING GOOD

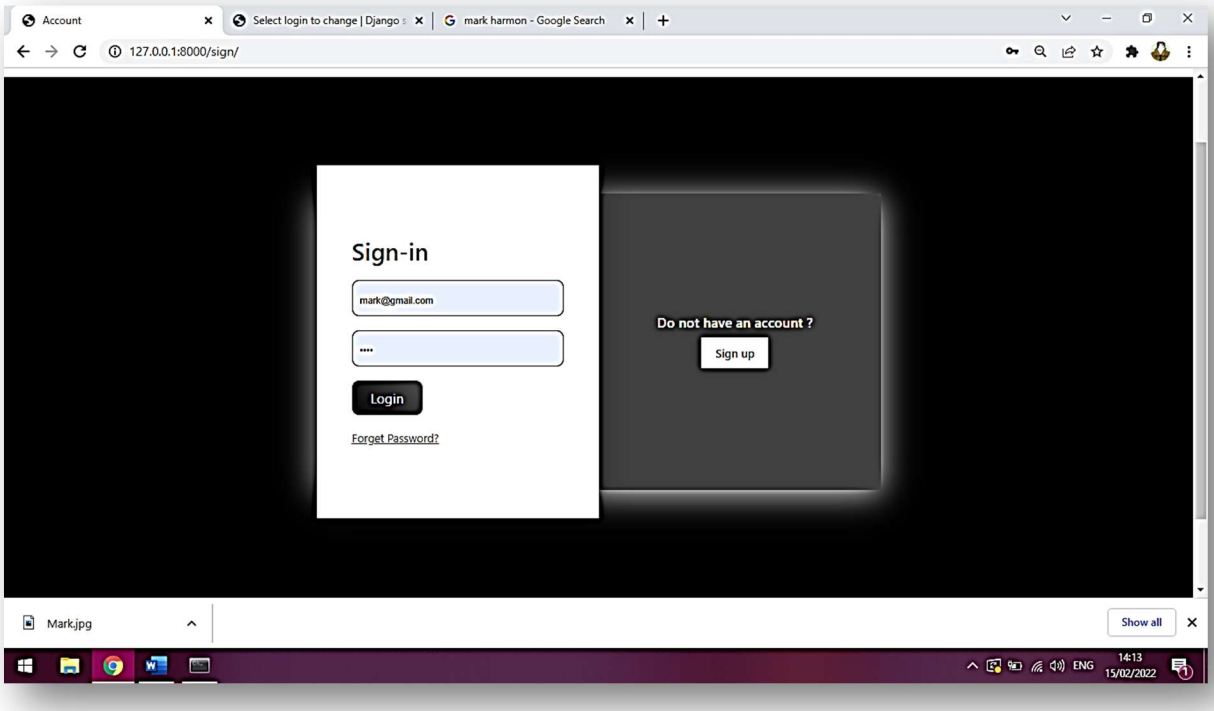


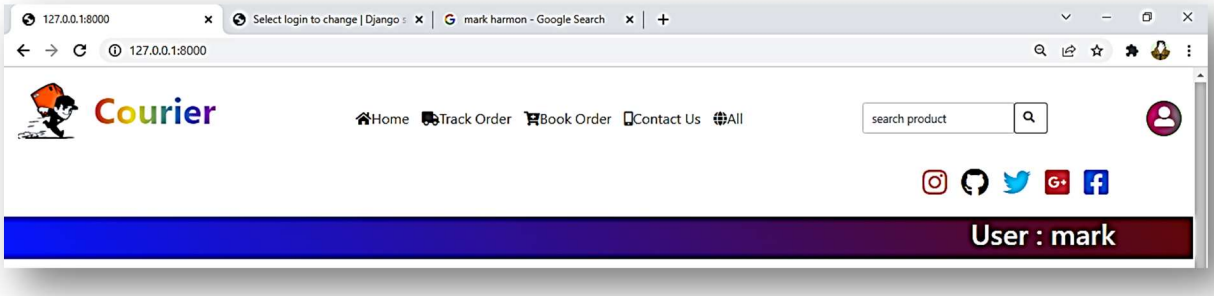


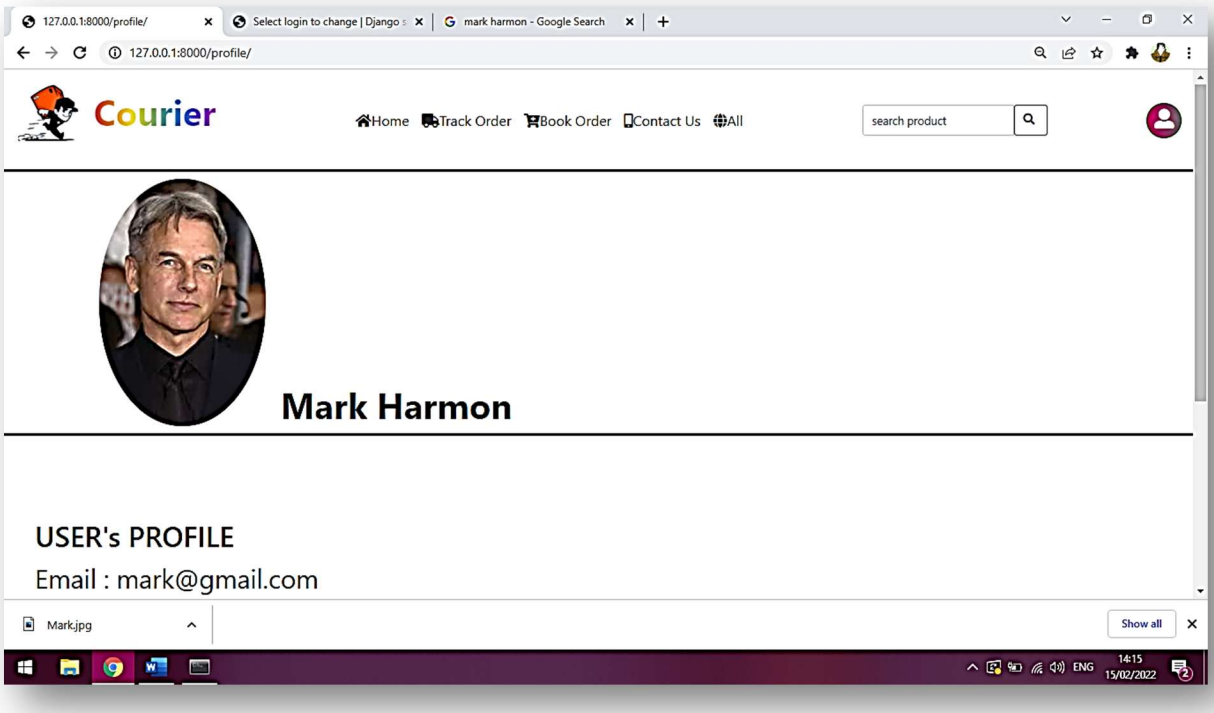
Compatibility Testing

Compatibility tests ensures that your web application displays correctly across different devices.

TESTING WEBSITE ON THE WINDOWS PLATFORM







TESTING RESULT :

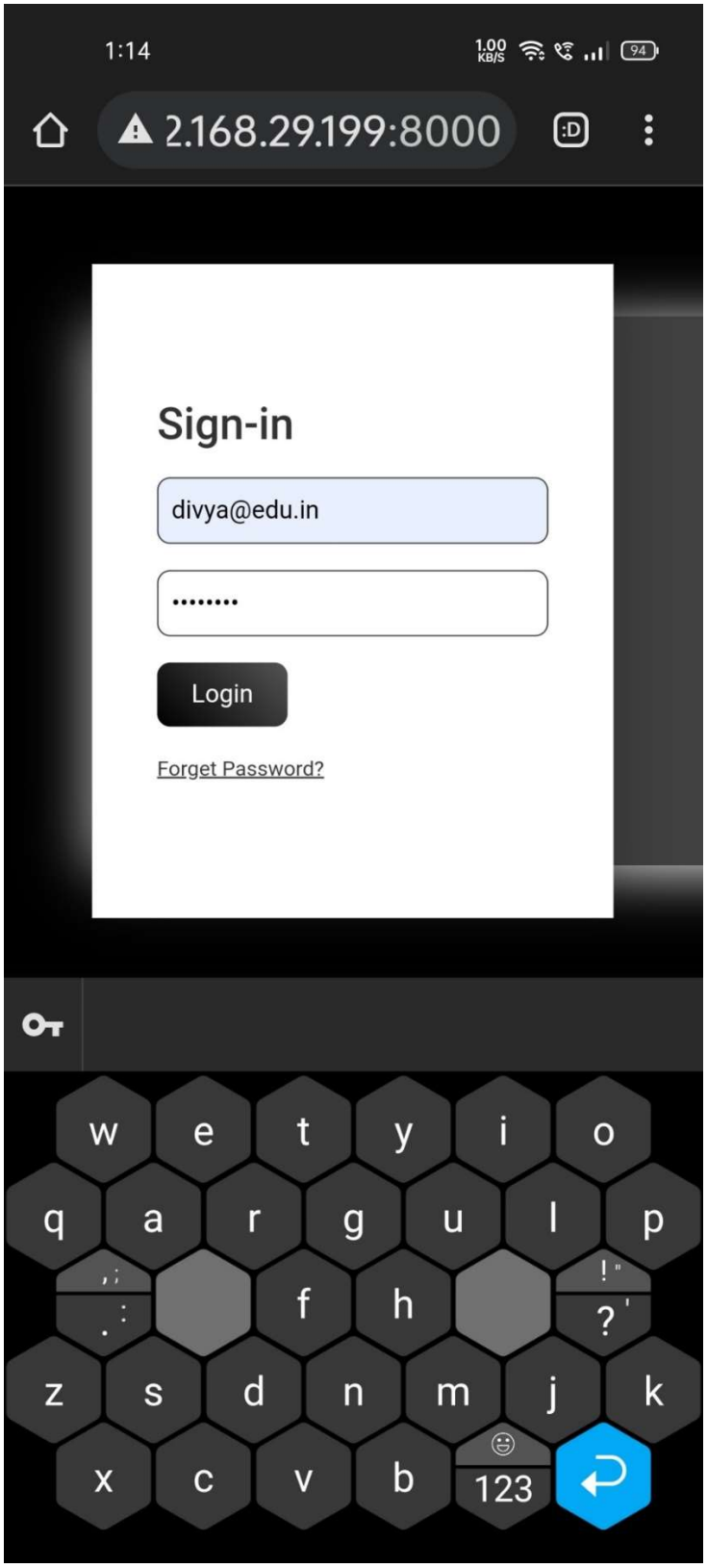
The website is working properly on windows platform.

All buttons are working.

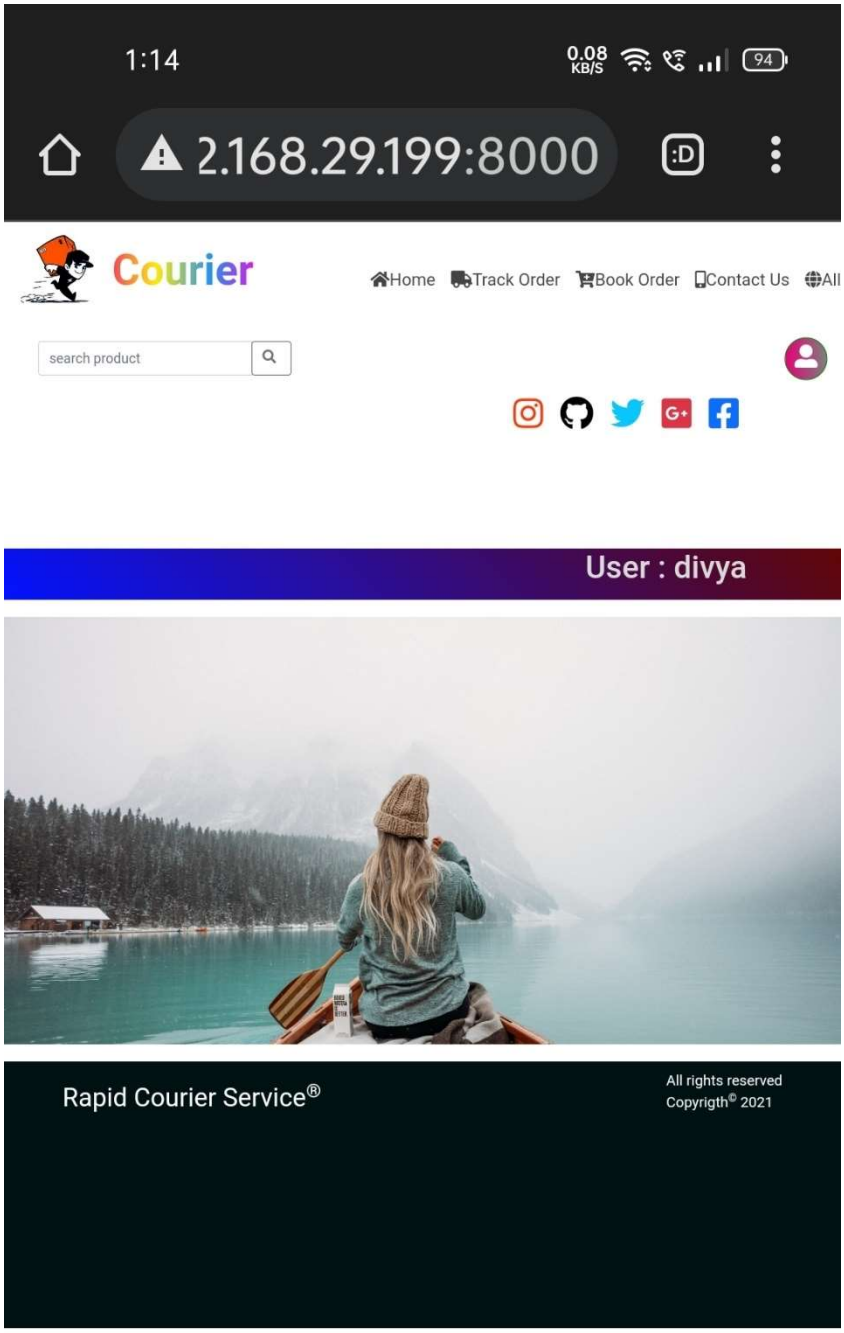
Contents are displayed as they are set to.

No irregularity found.

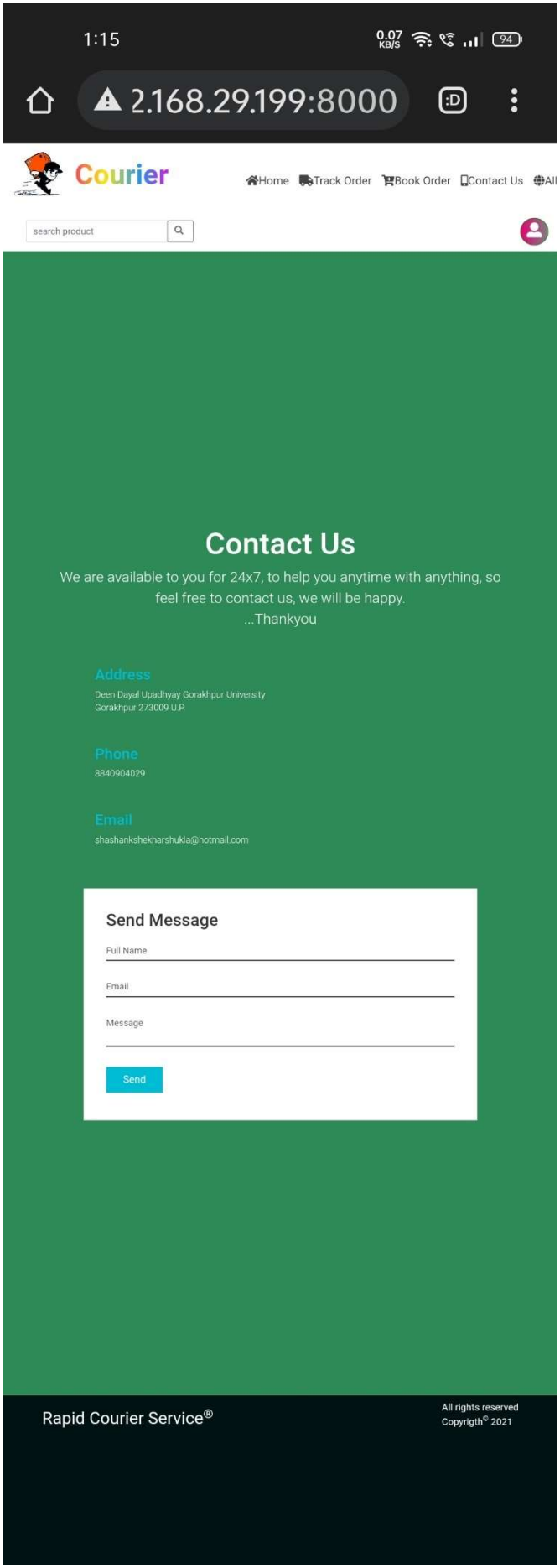
TESTING WEBSITE ON ANDROID PLATFORM

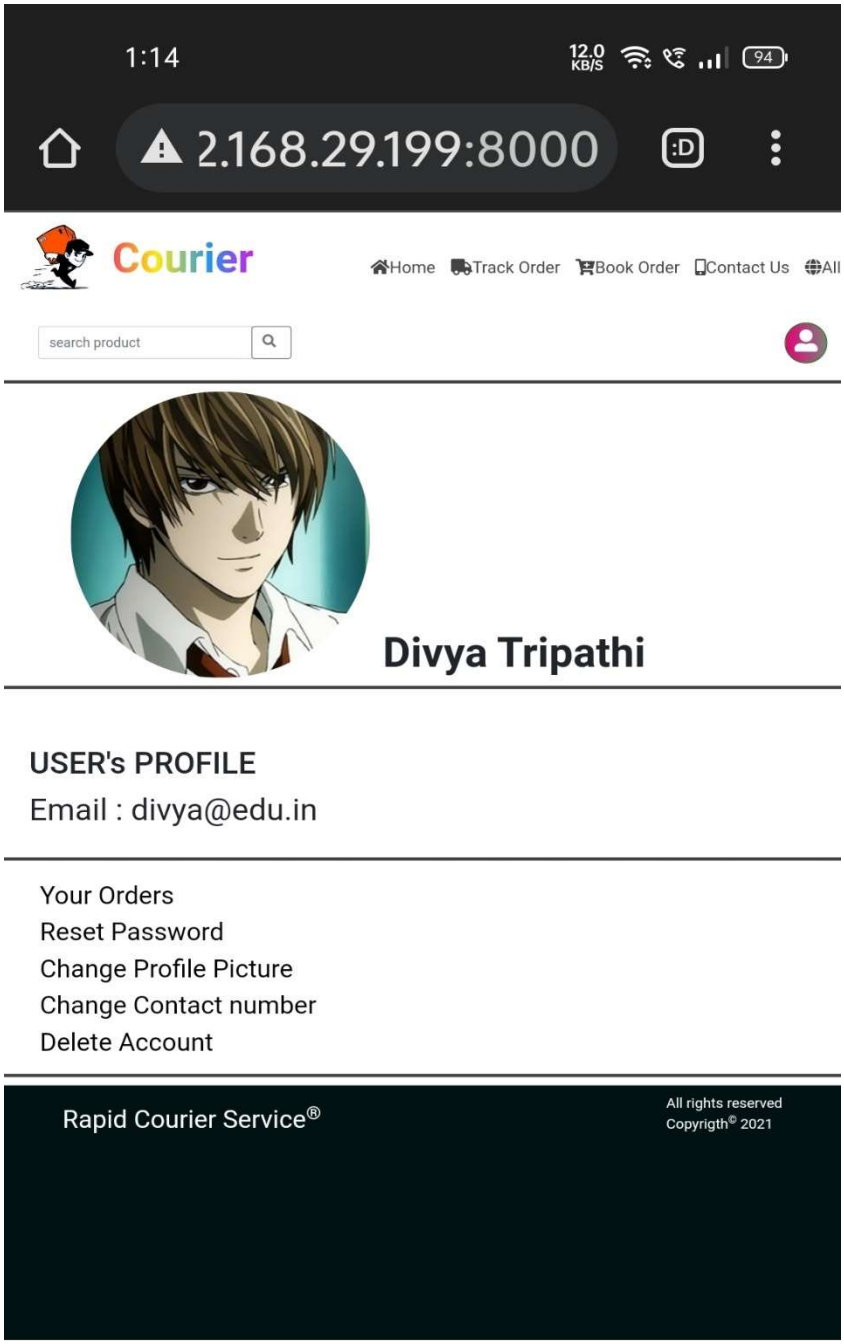


LOGGING IN

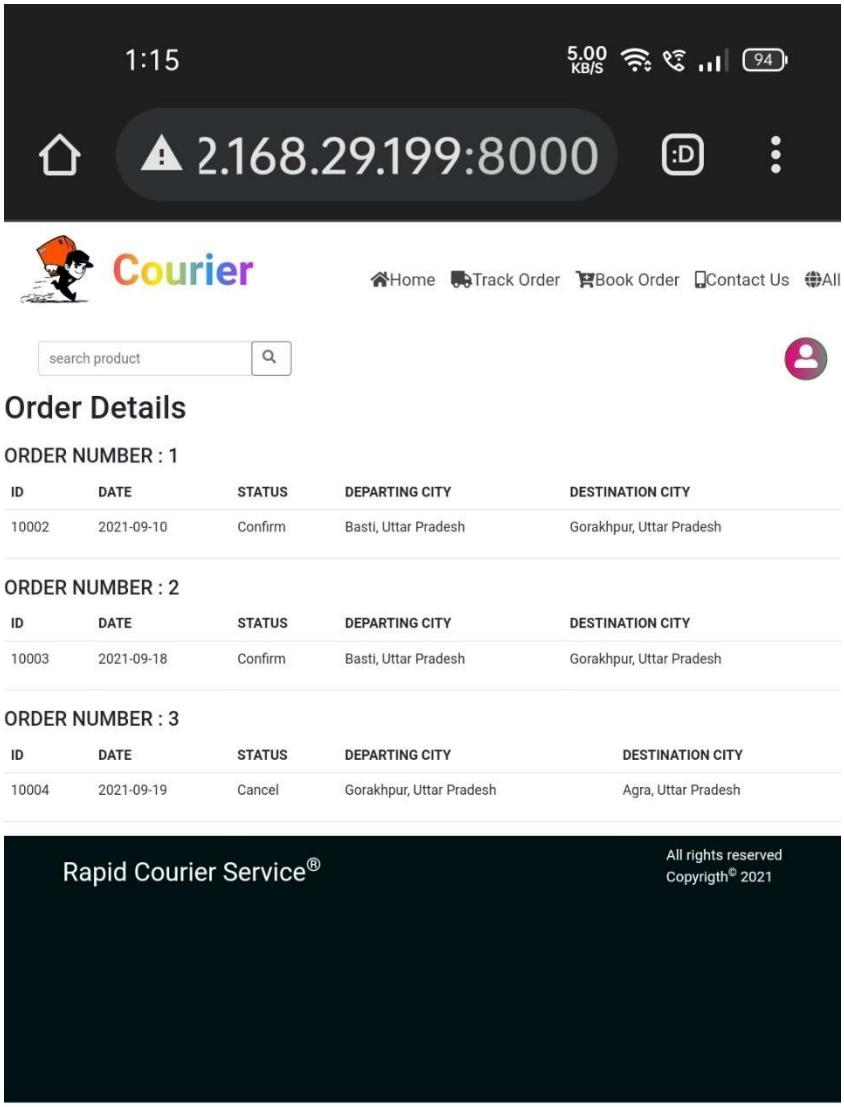


LOGGED IN SUCCESSFULLY


1:15
0.07 KB/S
2.168.29.199:8000
Courier
Home
Track Order
Book Order
Contact Us
All
search product
Contact Us
We are available to you for 24x7, to help you anytime with anything, so feel free to contact us, we will be happy.
...Thankyou
Address
Deen Dayal Upadhyay Gorakhpur University
Gorakhpur 273009 U.P
Phone
8840904029
Email
shashankshekharshukla@hotmail.com
Send Message
Full Name
Email
Message
Send
Rapid Courier Service®
All rights reserved
Copyrigth© 2021

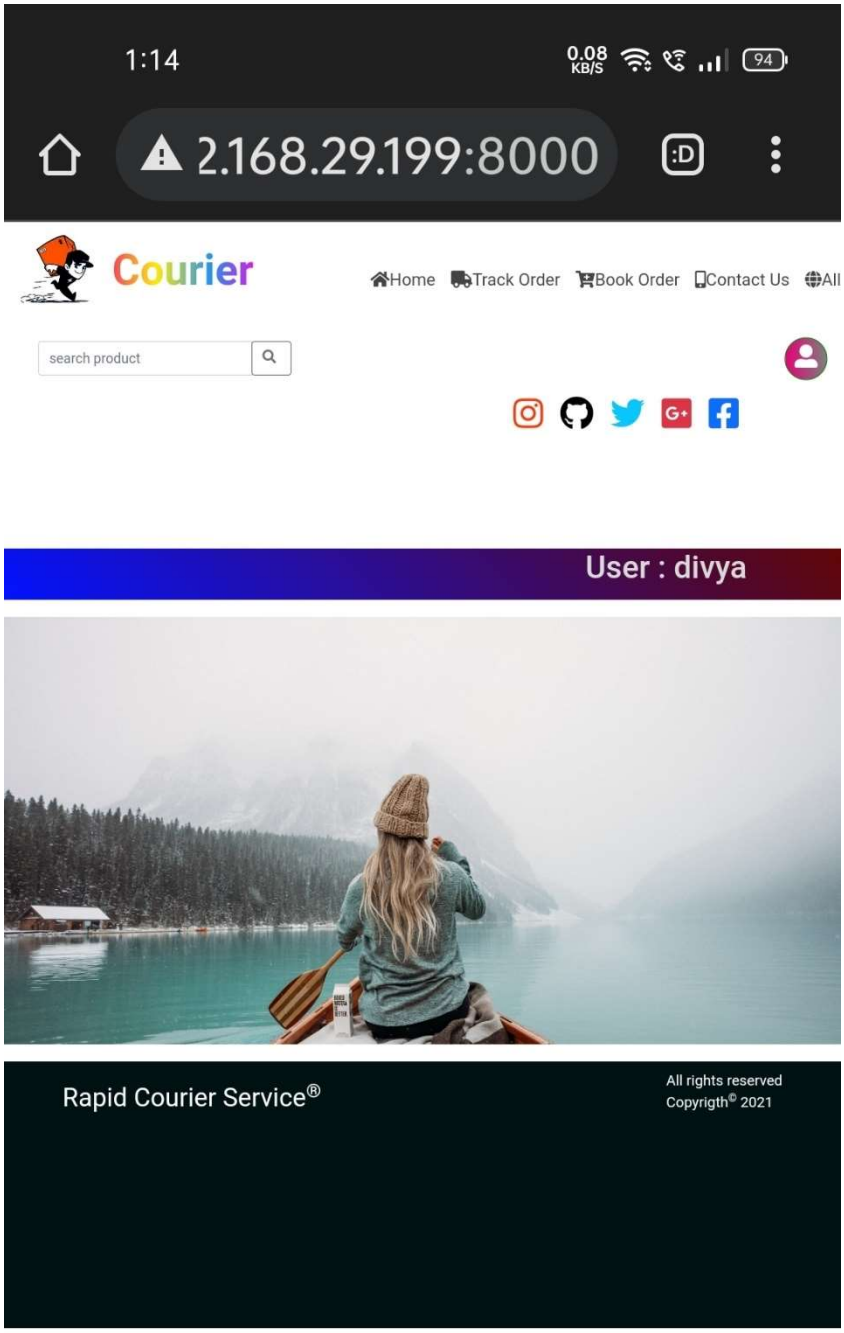


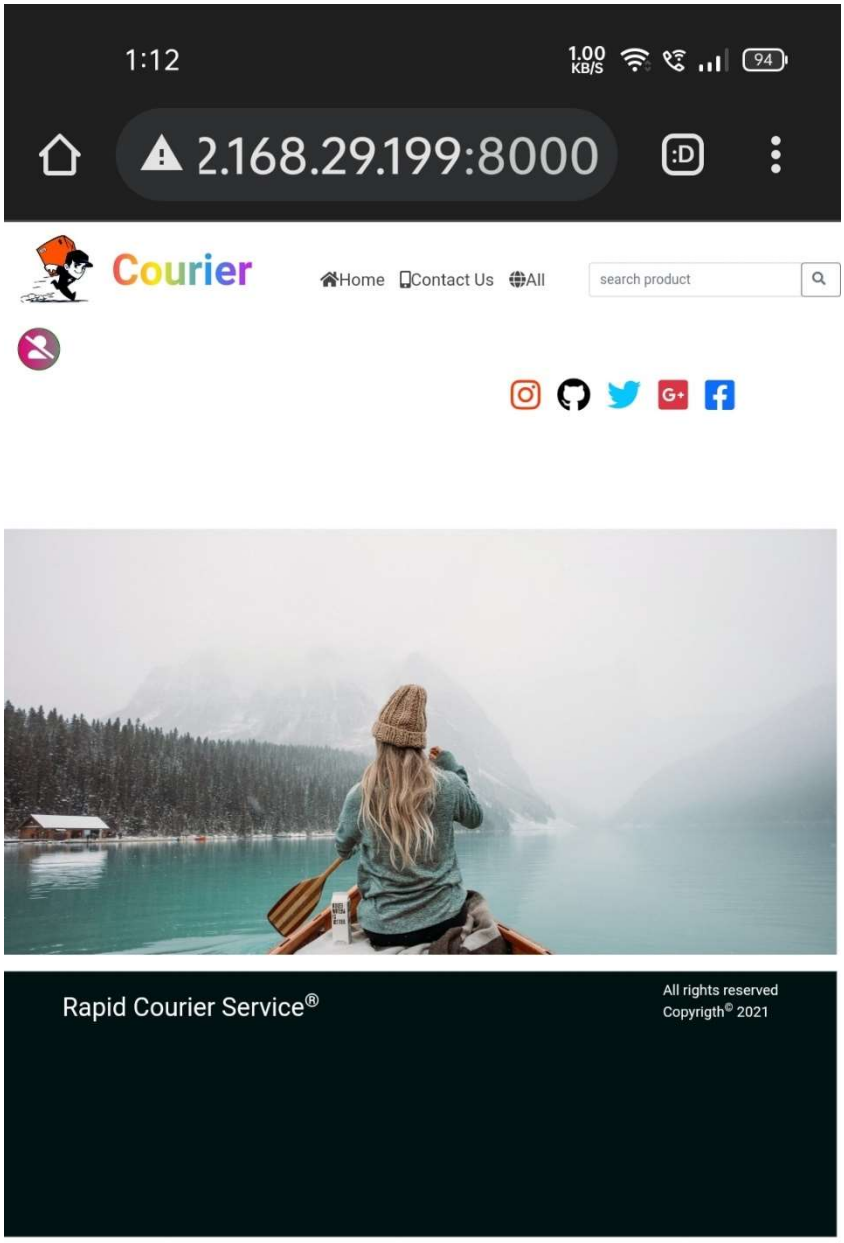
CHECKING PAGES OF THE WEBSITE



USER’s DATA IS SHOWN PROPERLY ON THE ANDROID DEVICE

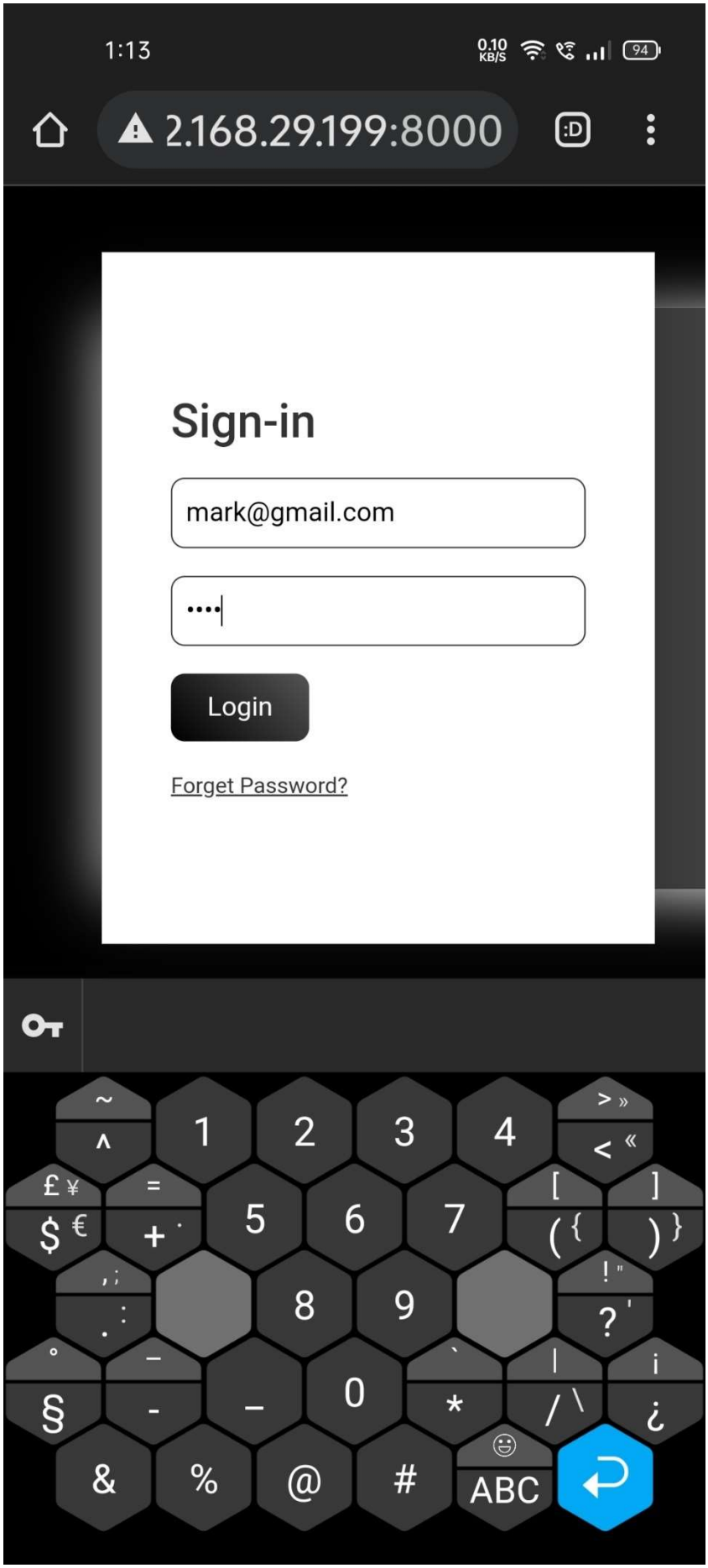
DATA SHOWN IN THE ABOVE TABLE BELONGS TO THE CURRENT USER [divya] AND IS 100% CORRECT

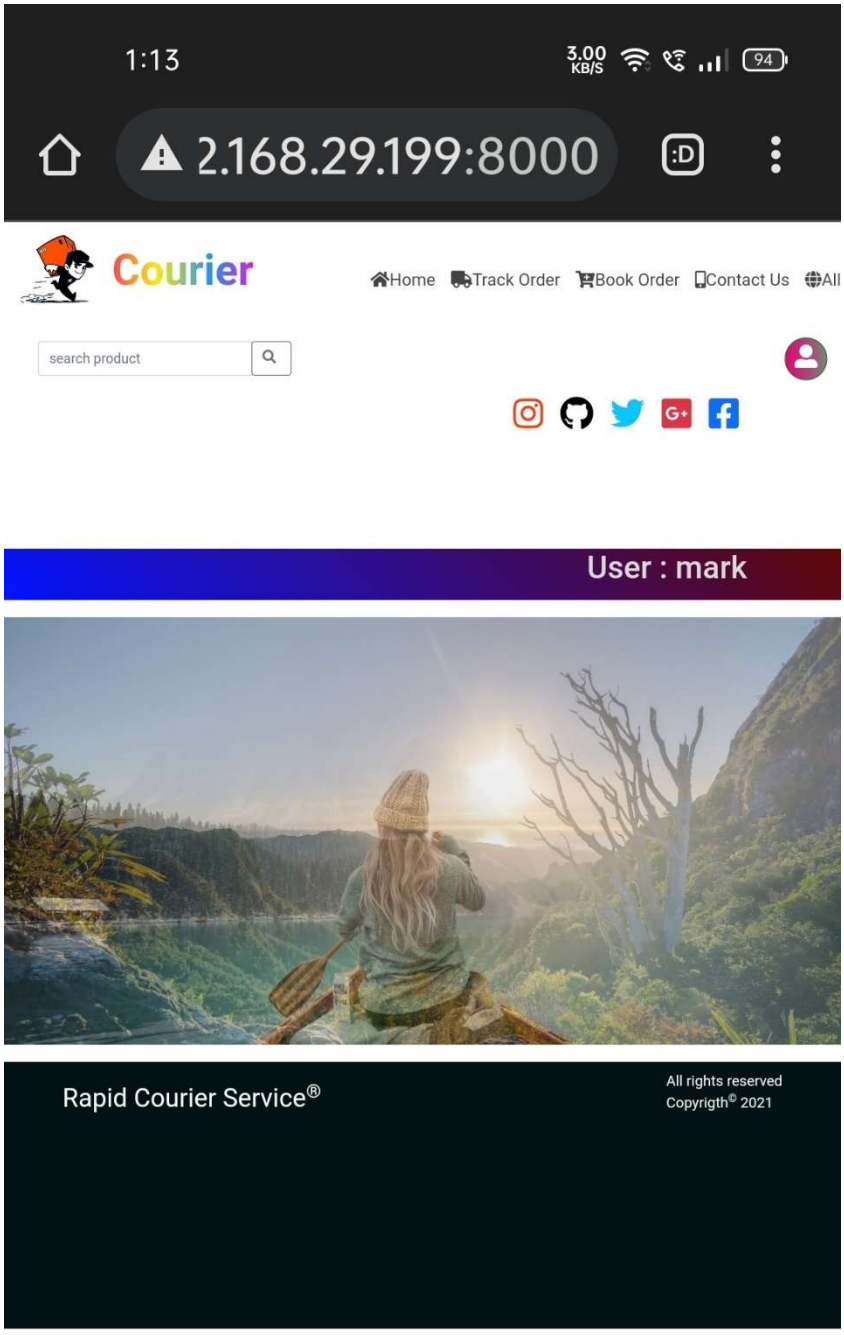

1:14
0.08 KB/S
94
2.168.29.199:8000
Courier
Home
Track Order
Book Order
Contact Us
All
search product
User : divya
Rapid Courier Service®
All rights reserved
Copyrigth© 2021



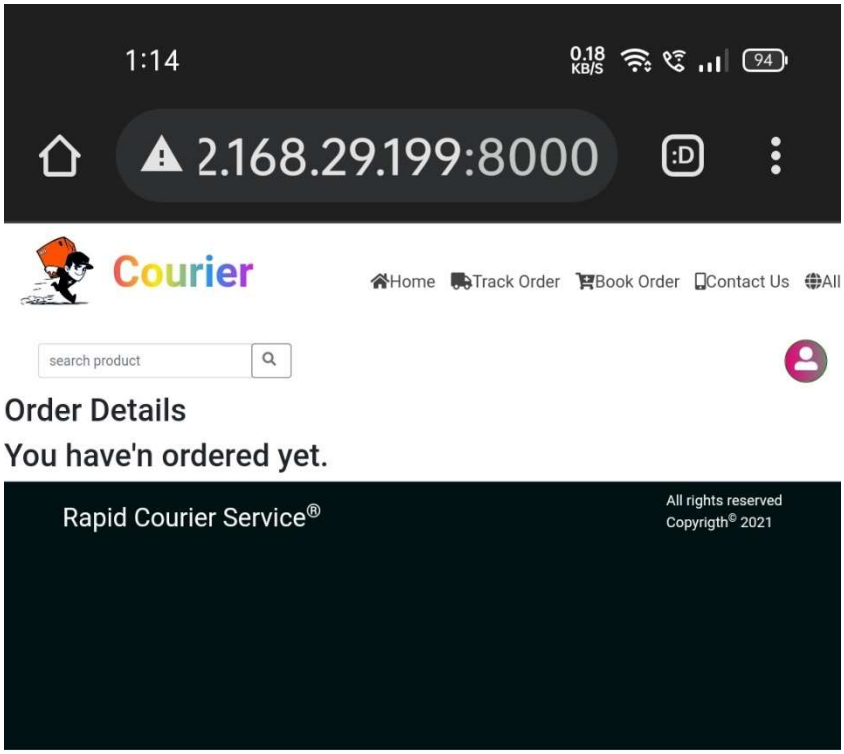
LOGGED OUT SUCCESSFULLY

Now trying to log-in with another type of user different from previous one


1:13
0.10 KB/S
94
2.168.29.199:8000
Sign-in
mark@gmail.com
Login
Forget Password?
ABC

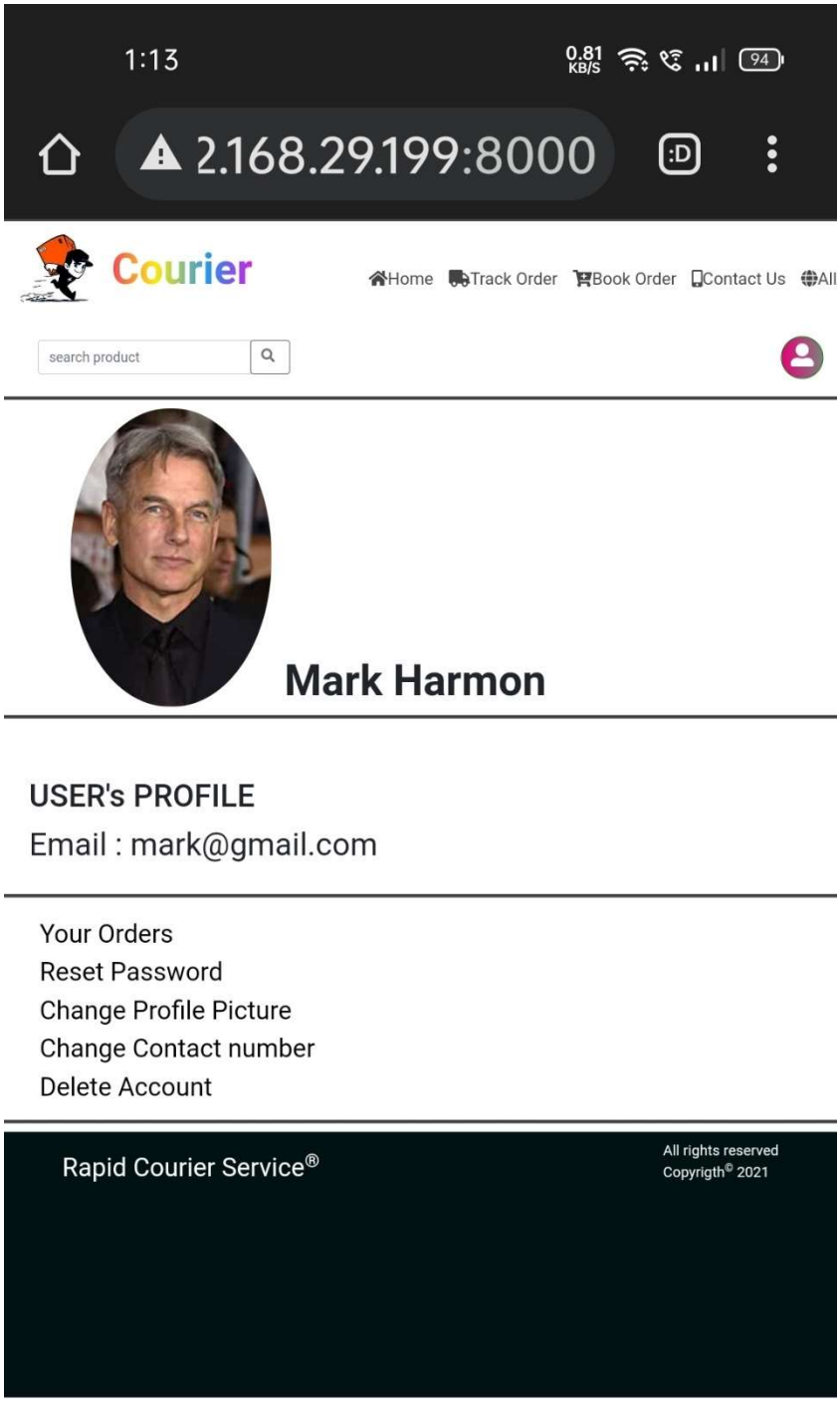


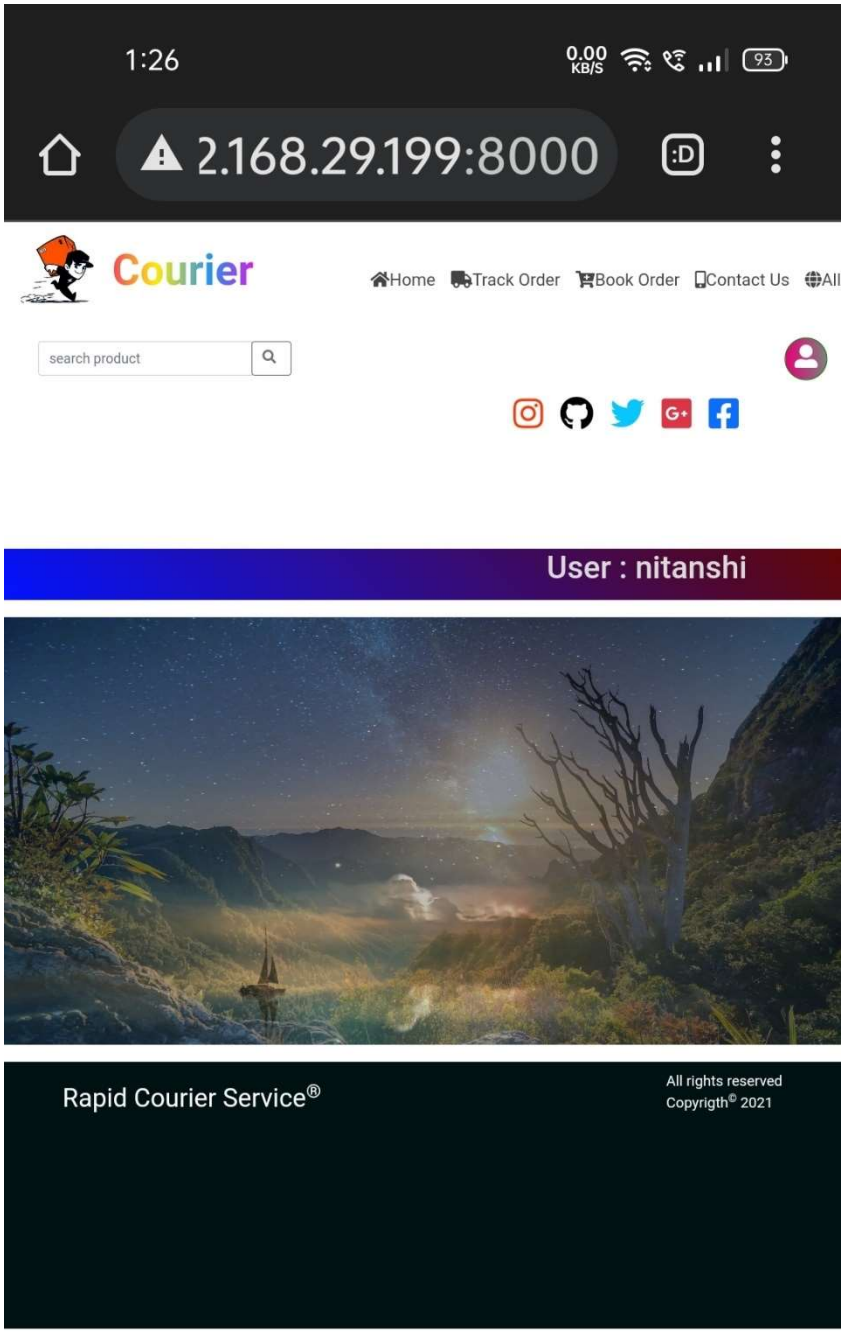
LOG-IN IS WORKING PROPERLY FOR ALL TYPES OF USERS

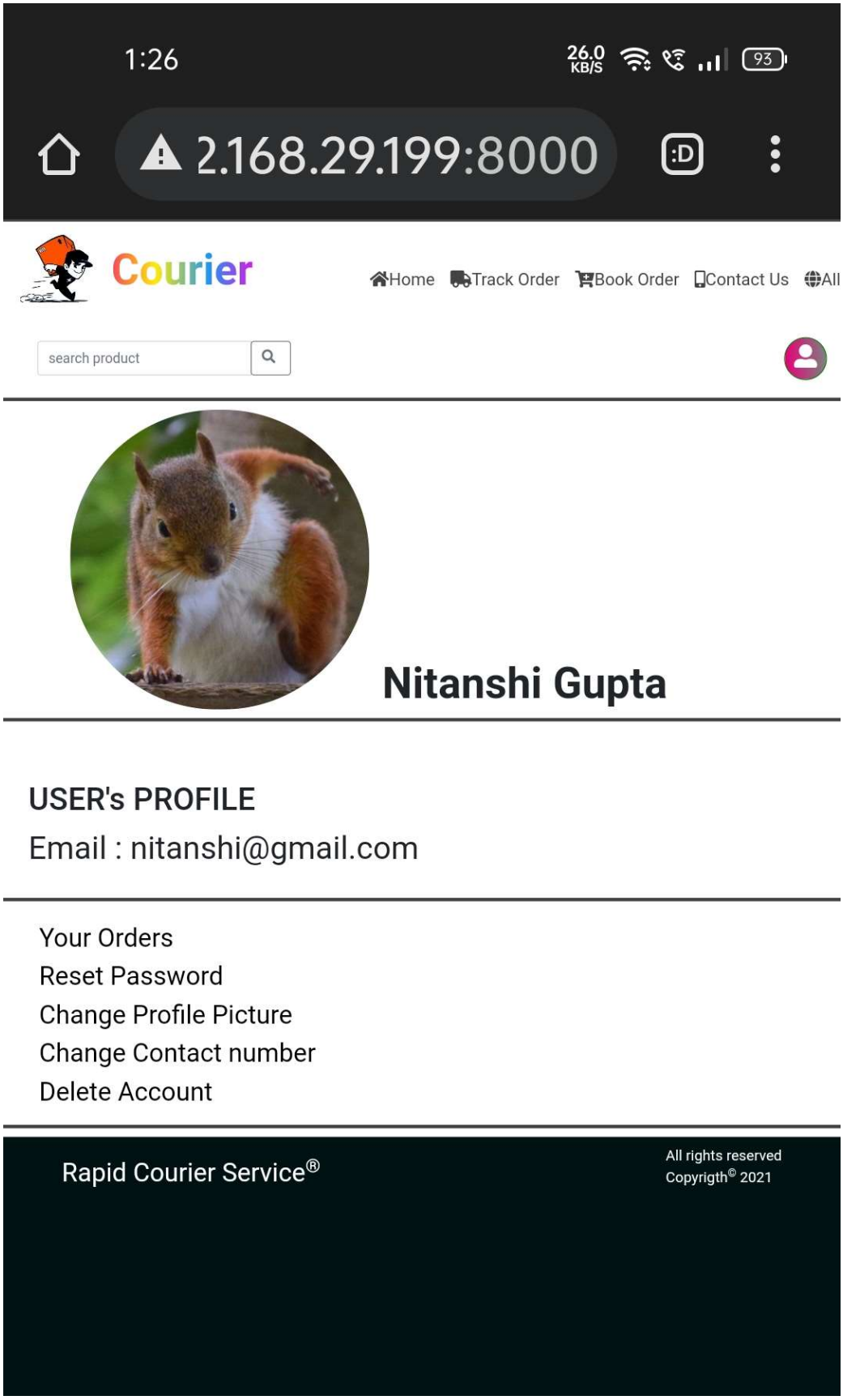


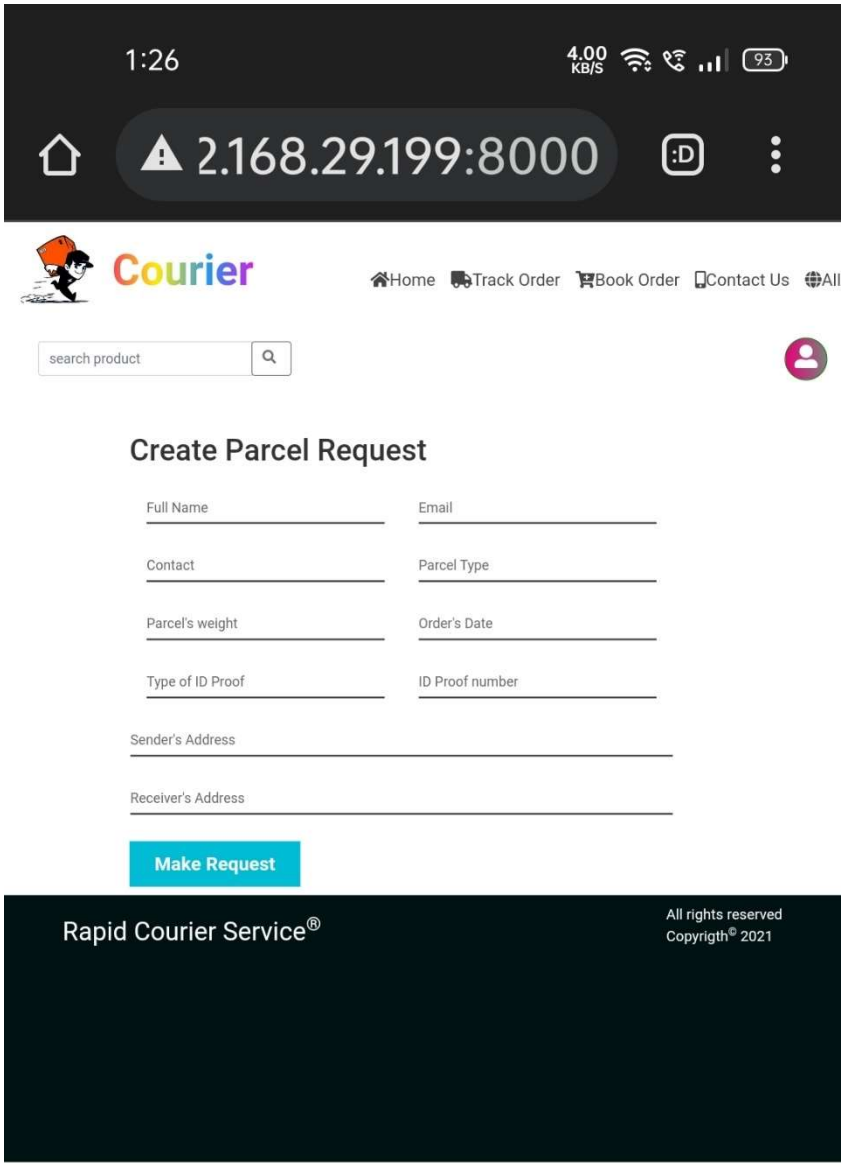
Data shown above is absolutely correct as this is a new user who hasn't made any order request yet.

This user has simple rights so is has restricted navigation over the web-pages.


1:13
0.81 KB/S
94
2.168.29.199:8000
Courier
Home
Track Order
Book Order
Contact Us
All
search product
Mark Harmon
USER's PROFILE
Email : mark@gmail.com
Your Orders
Reset Password
Change Profile Picture
Change Contact number
Delete Account
Rapid Courier Service®
All rights reserved
Copyrigth© 2021


1:26
0.00 KB/S
93
2.168.29.199:8000
Courier
Home
Track Order
Book Order
Contact Us
All
search product
User : nitanshi
Rapid Courier Service®
All rights reserved
Copyrigth© 2021


1:26
26.0 KB/S
93
2.168.29.199:8000
Courier
Home
Track Order
Book Order
Contact Us
All
search product
Nitanshi Gupta
USER's PROFILE
Email : nitanshi@gmail.com
Your Orders
Reset Password
Change Profile Picture
Change Contact number
Delete Account
Rapid Courier Service®
All rights reserved
Copyrigth© 2021



Security Testing

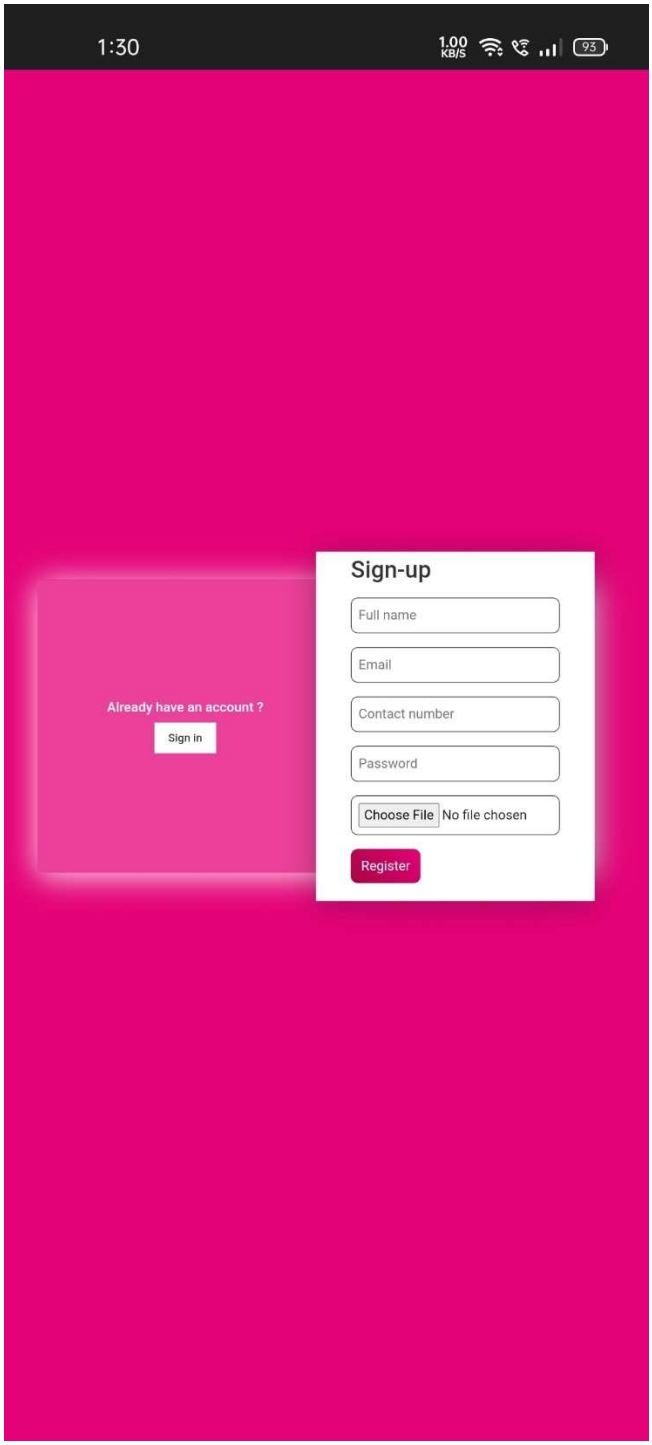
Under this,

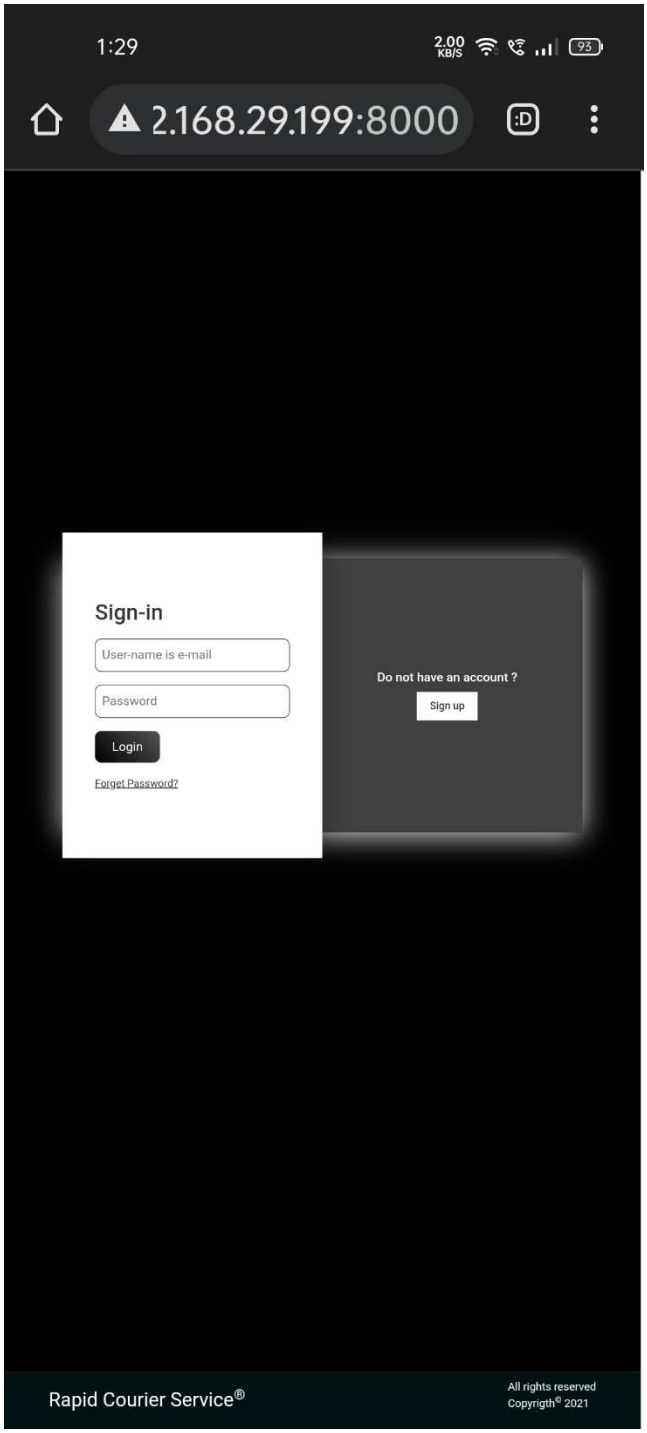
Tested the Unauthorised access to secure pages

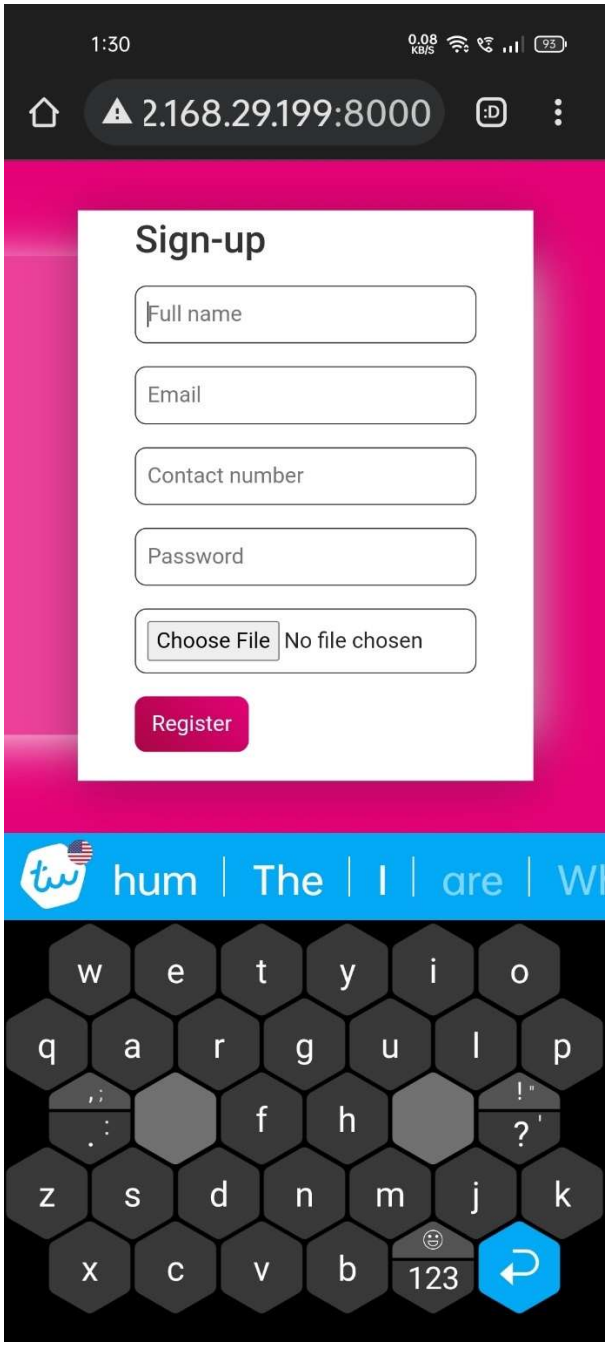
Tested the re-direction of web-pages by modifying html page

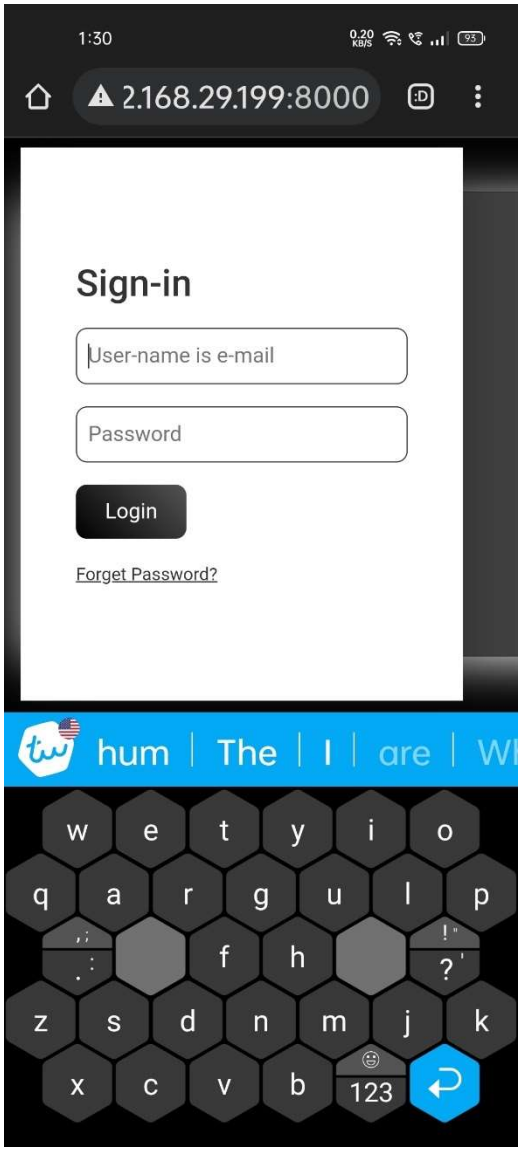
Tested SQL injection

Tested cross-site forgery


1:30
1.00 KB/S
93
Already have an account ?
Sign in
Sign-up
Full name
Email
Contact number
Password
Choose File
No file chosen
Register


1:29
2.00 KB/S
93
2.168.29.199:8000
Sign-in
User-name is e-mail
Password
Login
Forget Password?
Do not have an account ?
Sign up
Rapid Courier Service®
All rights reserved
Copyrigth© 2021


1:30
2.168.29.199:8000
Sign-up
Full name
Email
Contact number
Password
Choose File
No file chosen
Register
hum
The
I
are



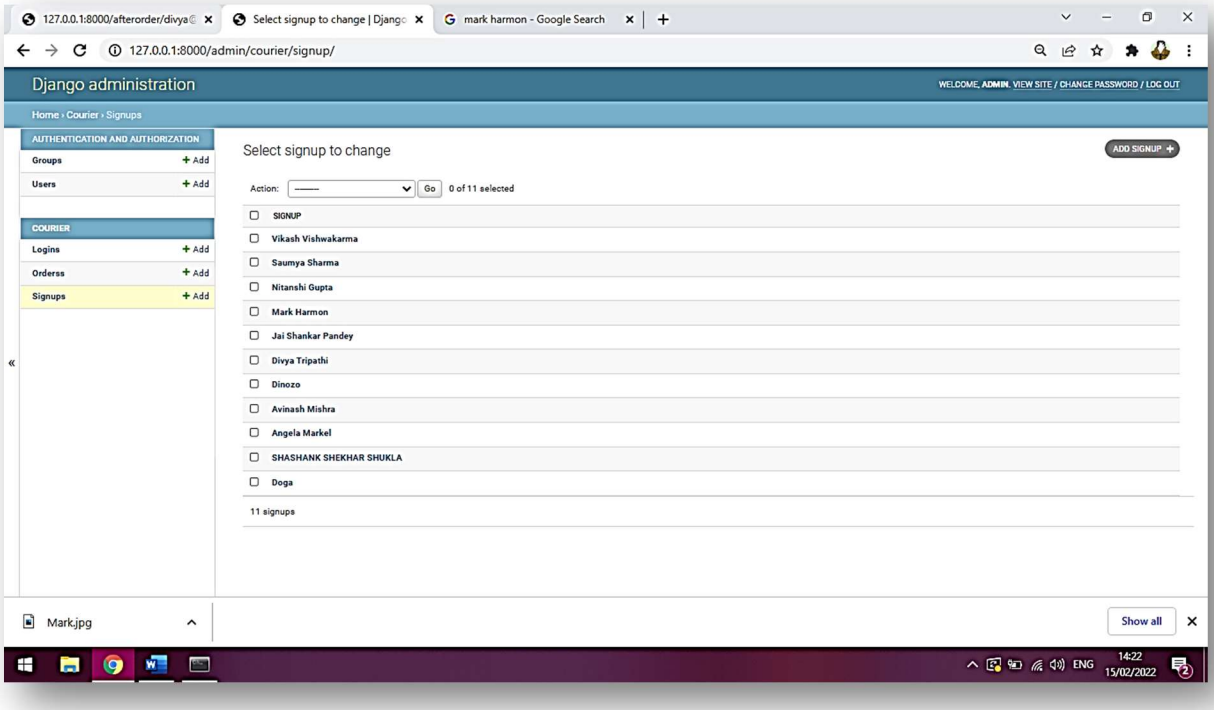
Unit Testing

Tested various modules developed individually

Then tested with all possible input, after satisfaction, I integrated modules with the core.

Integration Testing

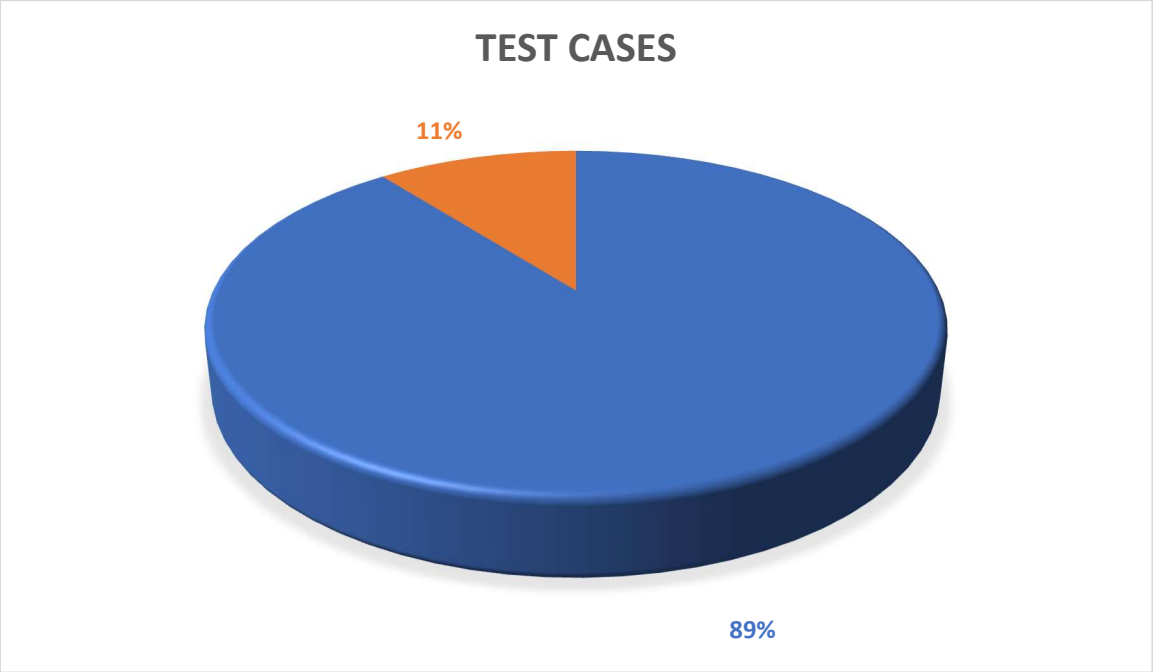
It is done to check whether the newly developed module is working fine when attached to the core or not.



All developed modules after passing the unit testing, got integrated into the core module, and the above database confirms that the integrated modules are working properly as these are designed to do so.

TESTING REPORT

TEST CASES			
PLANNED	EXECUTED	**PASSED**	**FAILED**
35	21	17	4



8 LIMITATIONS

1. Database used in this project is SqLite, which have less features.
2. SqLite have no recovery features.
3. Distance calculation is based on the abstract points measurement, not on the actual and correct distance.
4. Notification system is not yet applied into it.

9 FUTURE WORK

1. Enrich database features
2. Actual distance calculation
3. Notification system
4. Status of the courier

BIBILOGRAPHY

www.python.org

www.microsoft.docs

www.djangoproject.org

www.pycharm.org

The end,

Thank you.