

1. INTRODUCTION

1.1 Purpose of the Project

The purpose of project is to help users trade their used or new products online with ease. The users can upload details about their product and quote a price on the product and we'll find relevant buyers for them who are looking for that similar product. It can be used to overcome the problems faced by the users of finding cheap secondhand products. Here the sellers can be normal merchants or people who would want to sell their used products.

1.2 Motivation

Motivation for this project came from the need to improvise the existing websites that do not allow people except registered sellers to sell products. Also, these sellers sell only new products which are obviously not affordable to everyone. With our application we can strengthen the current e-commerce market.

1.3 Existing Procedure

Currently there are only a few websites that provide these services. It's very difficult to find an open e-commerce place to sell products.

2. PROJECT REQUIREMENTS

2.1 Hardware

- **Processor:** Intel i5 7th generation, 2.7 GHz, 64-bit
- **Ram:** 8GB
- **Hard disk:** 10 GB Free Space

2.2 Software

- **Operation System:** Windows 10/Mac OSX
- **Programming Language:** Node.js/Express.js
- **Database Language:** MySQL
- **Front-End Technology:** CSS, HTML, Bootstrap, JavaScript

3. LITERATURE SURVEY

3.1 Current Situation

A literature survey was conducted to create a theoretical background in the survey of product selling websites and applications. With the literature survey we aim to find the functionality that is used by the previous applications and to find additional functionality that may help and support this project.

Currently there are a few websites that provide these services. It's very difficult to find an open e-commerce place to sell used products. This becomes a huge trouble for people who have used products not being used by them anymore which they have to sell.

3.2 Similar Applications

- **OLX:** OLX Group is a global online marketplace, operating in 45 countries, and is the largest online classified ads company in Brazil, India, Bulgaria, Pakistan, Poland, Romania, Portugal and Ukraine. It was founded in 2006. Currently, olx.in is the most popular website in the market that provides features similar to ours. There are a few additional facilities OLX provides such as providing search results based on location of the user and features to chat with the seller.
- **Quikr:** Quikr is an Indian classified advertising platform. It was founded by Pranay Chulet and Jiby Thomas in 2008. Headquartered in Bangalore, Quikr has listings in over 900 cities in India in categories such as mobile phones, household goods, cars, real estate, jobs, services and education. It is also a similar website that which brings sellers to users who can sell their used items over the Internet.
- **eBay:** eBay Inc. is an American multinational e-commerce corporation based in San Jose, California that facilitates consumer-to-consumer and business-to- consumer sales through its website. eBay was founded by Pierre Omidyar in Fall 1995, and became a notable success story of the dot-com bubble.

4. PROBLEM STATEMENT

The problem arises from the people's need for a trusted classified website. A classified website is a place where people (not necessary companies) can buy / sell things and they are categorized by:

Type:

- Job Boards
- Second hand cars
- Real Estate websites
- Second Hand items

Location:

- By countries
- States
- Cities
- Towns

Categories:

- Jobs for IT industry or Media Industry
- Second Hand Cars, just 4x4 or SUVs
- Real Estate Websites for Luxury only.
- Second Hand items only furniture, or clothing.

5. SYSTEM DESIGN

Systems design is the process of defining the architecture, modules, interfaces, and data for a system to satisfy specified requirements. Systems design could be seen as the application of systems theory to product development. There is some overlap with the disciplines of systems analysis, systems architecture and systems engineering.

5.1 Dataflow Diagram

In our project which is a graphical representation of the "flow" of data through an information system, modelling its *process* aspects. A DFD is often used as a preliminary step to create an overview of the system without going into great detail, which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design).

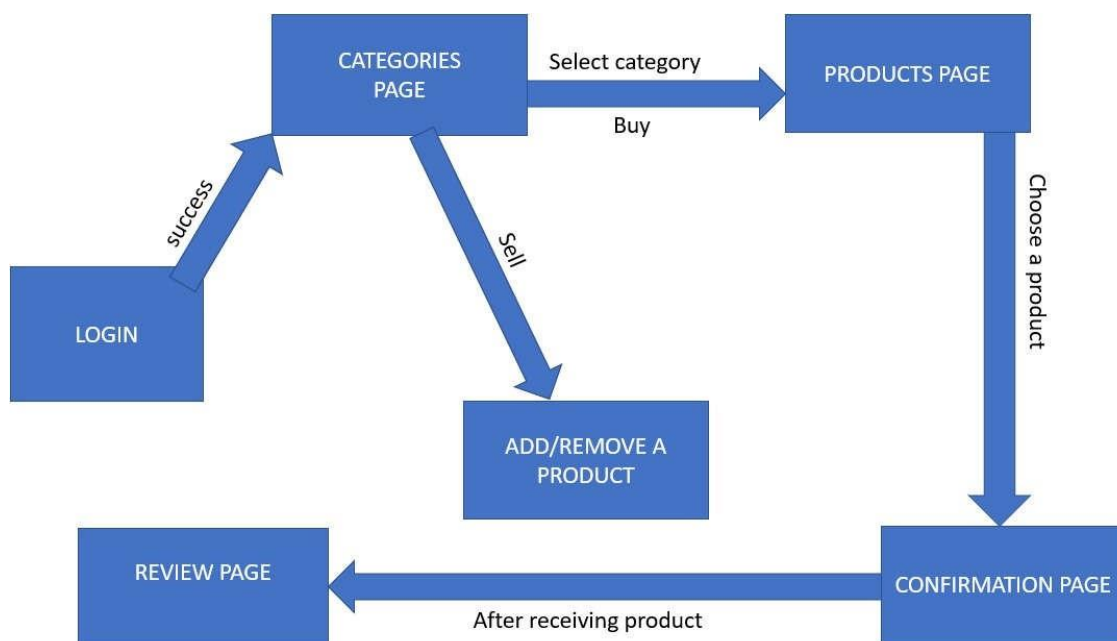


Fig 5.1 Dataflow Diagram

6. IMPLEMENTATION

6.1 Frontend Technology

6.1.1 JavaScript

- **Node.js:** We've used Node.js to run our server which hosts our web application. This includes all the HTML, CSS and JavaScript files. Node.js is the most popular server-side scripting language currently being used in majority of the servers.
- **Express.js:** Express is a Node.js web application framework that provides robust set of features. It supports a lot of popular view engines and middleware.

6.1.2 HTML

HTML5 is the latest version of Hypertext Mark-up Language that is used to define the structure of web pages. Hypertext Mark-up Language is the standard mark-up language for creating web pages and web applications.

6.1.3 CSS 3

CSS3 is the latest version of the CSS specification. CSS3 adds several new styling features and improvements to enhance the web presentation capabilities. CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts.

6.1.4 Bootstrap 4

Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains CSS- and JavaScript-based design templates for typography, forms, buttons, navigation and other interface components.

6.2 Backend Technology

6.2.1 Node.js Code Snippets

```
const express = require('express');
const mysql = require('mysql');
const port = 3031;

var app = express();

// Static directory: public
app.use(express.static(__dirname+"/public"));
app.set('view engine', 'ejs');

// MySQL connection setup
var conn = mysql.createConnection({
  connectionLimit: 50,
  host: "localhost",
  user: "parth",
  password: "password",
  database: "ecommerce"
});

conn.connect((err) => {
  if (err) {
    console.log(err);
  } else {
    console.log("Connection to MySQL server successful.");
  }
});

// Home Page and All Products Page
app.get("/", (req, res) => {
  console.log("GET " + req.url);
  var sql_query = "SELECT * FROM products WHERE status=1";
  conn.query(sql_query, (err, rows, fields) => {
    if (err) {
      console.log(err);
    } else {
      res.render("landing.ejs", {
```

```
                products: rows
            });
        }
    });
});

// Orders Page
app.get("/orders", (req, res) => {
    console.log("GET " + req.url);
    res.render("orders.ejs");
});

//Auth routes
app.get("/register",function(req,res){
    res.render("register.ejs");
});

app.post("/register",function(req,res){
    var newUser=new User({username: req.body.username});
    //register this user using passport
    User.register(newUser,req.body.password,function(err,user){
        if(err){
            console.log(err);
            return res.render("register.ejs");
        }
        passport.authenticate("local")(req,res,function(){
            res.render("user_info.ejs");
        });
    });
});

//Login routes
app.get("/login",function(req,res){
    res.render("login.ejs");
});

app.post("/login",passport.authenticate("local",{
    successRedirect:"/products",
    failureRedirect:"/login"
}), function(req,res){
```



```
    res.render("Login logic happens here");
  });

//logout routes
app.get("/logout",function(req,res){
  req.logout();
  res.redirect("/products");
});

function isLoggedIn(req,res,next){
  if(req.isAuthenticated()){
    return next();
  }
  res.redirect("/login");
}

app.listen(3000||process.env.PORT,function(){
  console.log("App running at 3000");
});
```

6.2.2 HTML Code Snippets

```
<body>
  <div class="container" style="padding-top: 50px;">
    <!-- navbar -->
    <nav class="navbar navbar-light bg-light navbar-expand-
sm fixed-top">
      <a href="/" class="navbar-brand">Trade It Off</a>
      <button class="navbar-toggler" data-toggle="collapse" data-
target="#navbarCollapse">
        <span class="navbar-toggler-icon"> </span>
      </button>
      <div class="collapse navbar-collapse" id="navbarCollapse">
        <ul class="navbar-nav ml-auto">

          <li class="navbar-item">
            <div class="dropdown">
              <button class="btn dropdown-
toggle" type="button" data-toggle="dropdown">Categories
              <span class="caret"></span></button>
```

```
        <ul class="dropdown-menu">
            <li><a href="/mobiles"> Mobiles </a></li>
            <li><a href="/furniture"> Furniture </a></li>
            <li><a href="/bikes"> Bikes </a></li>
        </ul>
    </div>
</li>

<li class="navbar-item">
    <a href="/orders" class="nav-link"> Orders </a>
</li>

<li class="navbar-item">
    <a href="/login" class="nav-link"> Login </a>
</li>

<li class="navbar-item">
    <a href="/signup" class="nav-link"> Sign Up </a>
</li>
</ul>
</div>
</nav>
</div>

//login.ejs

<html>
<script>
    addEventListener("load", function () {
        setTimeout(hideURLbar, 0);
    }, false);

    function hideURLbar() {
        window.scrollTo(0, 1);
    }
</script>

<!-- Custom Theme files -->
<link          href="/stylesheets/style.css"          rel="stylesheet"
type="text/css" media="all" />
```

```
<link href="/stylesheets/font-awesome.min.css" rel="stylesheet"
type="text/css" media="all" />
<!-- //Custom Theme files -->

<!-- web font -->
<link
href="//fonts.googleapis.com/css?family=Hind:300,400,500,600,700"
rel="stylesheet">
<!-- //web font -->
<body>

<!-- main -->
<div class="w3layouts-main">
  <div class="bg-layer">
    <h1>Welcome Back!</h1>
    <div class="header-main">
      <div class="main-icon">
        <span class="fa fa-eercast"></span>
      </div>
      <div class="header-left-bottom">
        <form action="/login" method="post">
          <div class="icon1">
            <span class="fa fa-user"></span>
            <input type="email" name="email"
placeholder="Email Address" required="" />
          </div>
          <div class="icon1">
            <span class="fa fa-lock"></span>
            <input type="password"
name="password" placeholder="Password" required="" />
          </div>
          <div class="login-check">
            <label class="checkbox"><input
type="checkbox" name="checkbox" checked=""><i> </i> Keep me logged
in</label>
          </div>
          <div class="bottom">
            <button class="btn">Log In</button>
          </div>
          <div class="links">
```

```

                                <p><a                                href="#">Forgot
Password?</a></p>
                                <p    class="right"><a    href="#">New
User? Register</a></p>
                                <div class="clear"></div>
                                </div>
                                </form>
                                </div>
                                <div class="social">
                                <ul>
                                <li>or login using : </li>
                                <li><a    href="#"    class="facebook"><span
class="fa fa-facebook"></span></a></li>
                                <li><a    href="#"    class="twitter"><span
class="fa fa-twitter"></span></a></li>
                                <li><a    href="#"    class="google"><span
class="fa fa-google-plus"></span></a></li>
                                </ul>
                                </div>
                                </div>
                                </div>
                                </div>
                                <!-- //main -->

</body>
</html>
```

6.3 Database

MySQL: We've used MySQL to implement our database. MySQL is an open source relational database management system (RDBMS) based on Structured Query Language (SQL). MySQL runs on virtually all platforms, including Linux, UNIX, and Windows.

7. TESTING

7.1 Unit Testing

The three units, namely, database unit, views unit and backend NODEJS unit are tested individually before integrating them into one single web application.

- The database is tested through a number of DDL (Data Definition Language) and DML (Data Manipulation Language) commands in order to discover inconsistencies that arise regarding prime key constraints and other referential integrity constants.
- The views unit is primarily tested for its frontend functionality that is, for the user interface. It is ensured that the input forms function as they are required. The HTML pages are tested individually to avoid any discrepancies in the layout, format and style of the user interface. After linking the HTML pages to the CSS files to obtain the required style for the web application and the JavaScript files to perform logic on the front end, tests are performed again in order to ensure no inconsistency while linking.
- The NODEJS unit is individually tested for proper database connectivity and to discover any errors that might arise while using the express mysql package. Queries are executed using different Statement objects with test data to ensure the right connection is established with the database and the queries yield required output.

After it is ensured that the individual unit work fine and generate the required output, they are integrated into a web application.

7.2 Integration Testing

After the three individual units have been integrated into a single web application, it is essential to check that the application works as a whole. The integration testing is started by testing that all the web pages can be accessed. Smooth transitions from one web page to the correct redirected page is also checked. The forms are tested with

input test data and verified that the input is received and processed by the server program that has established a connection with the database. The test data received as input is used to modify the database through the express –

MySQL API and SQL commands. This procedure is also tested to ensure correct access, insertion and update of the tables corresponding to the action performed by the user.

7.3 Integration Results

Test ID.	Test Case Condition	Input	Expected Output
01	When user clicks Register button.	Register (Onclick)	Register window
02	When user click on Feedback button.	Onclick feedback()	Feedback window
03	When admin clicks on login button.	Onclick feedback()	User Details Page
04	When admin clicks on Admin-Login button.	Onclick admin-login()	Admin-login window drops down.
05	When user clicks on Add Product button.	Onclick addProduct(form)	Product details page to describe the product.
06	When user clicks on show all products button	Onclick allProducts()	Products from all the different categories page shows up

Table 7.1 Testing Results

8. RESULTS

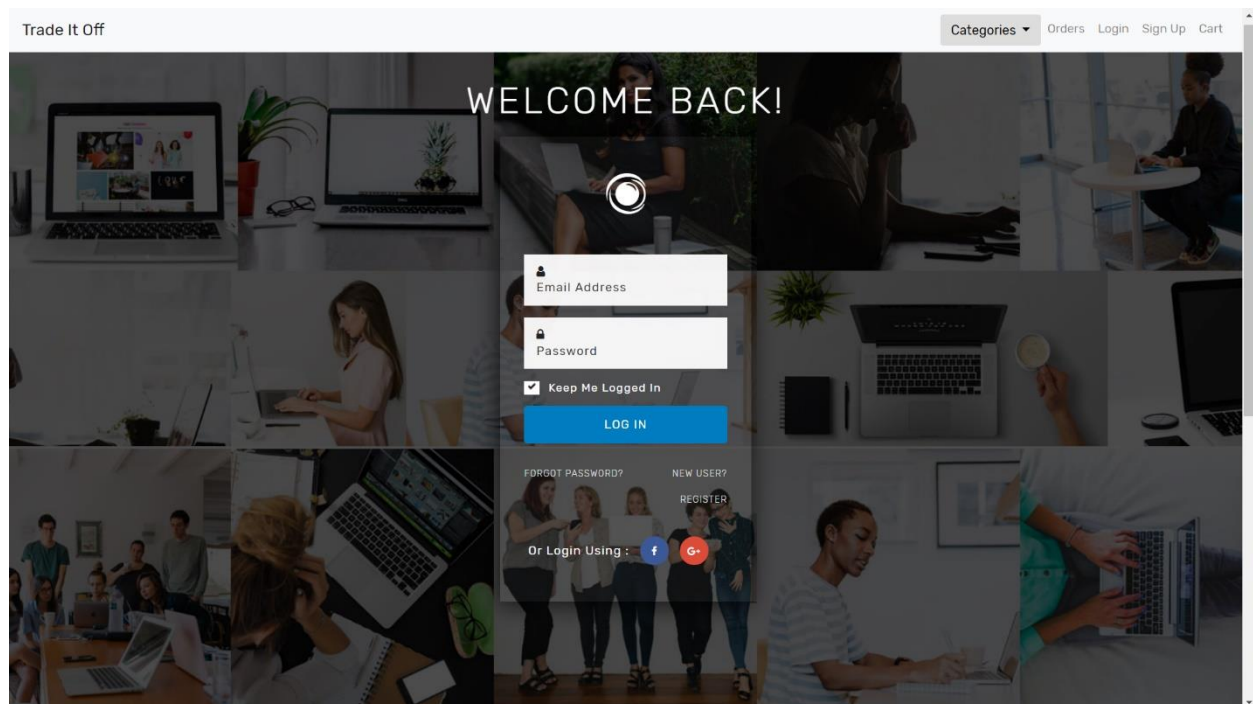


Fig 8.1 Login Page

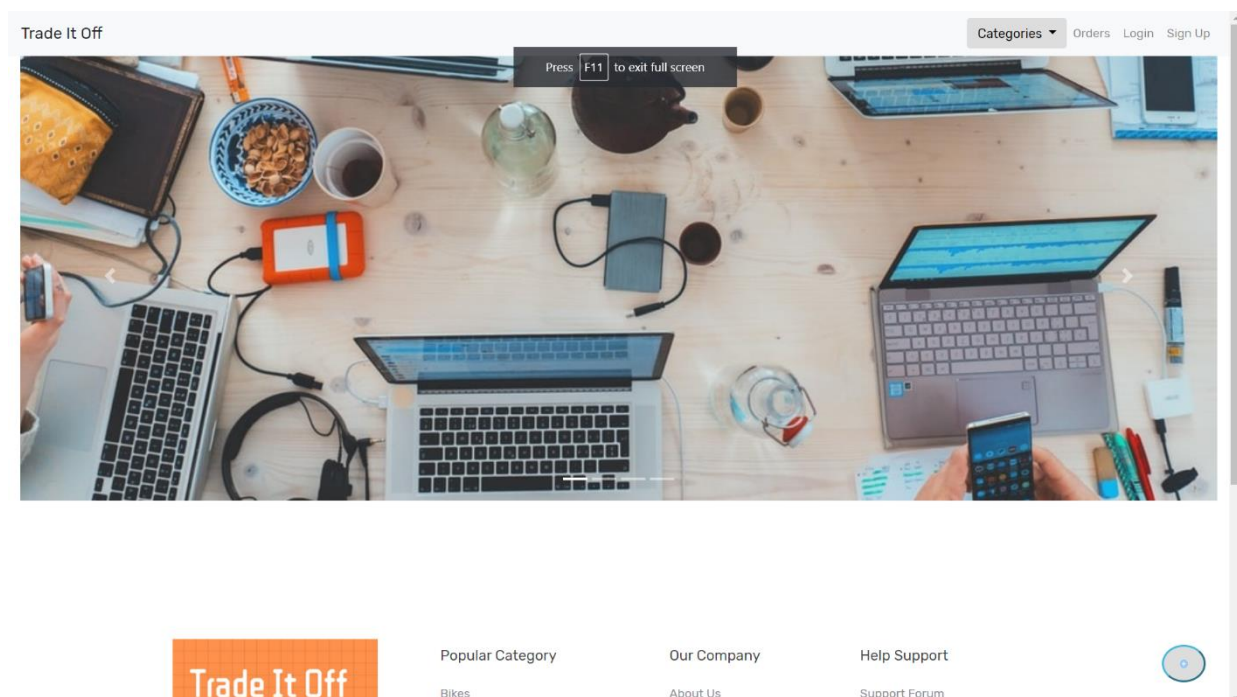


Fig 8.2 Home Page

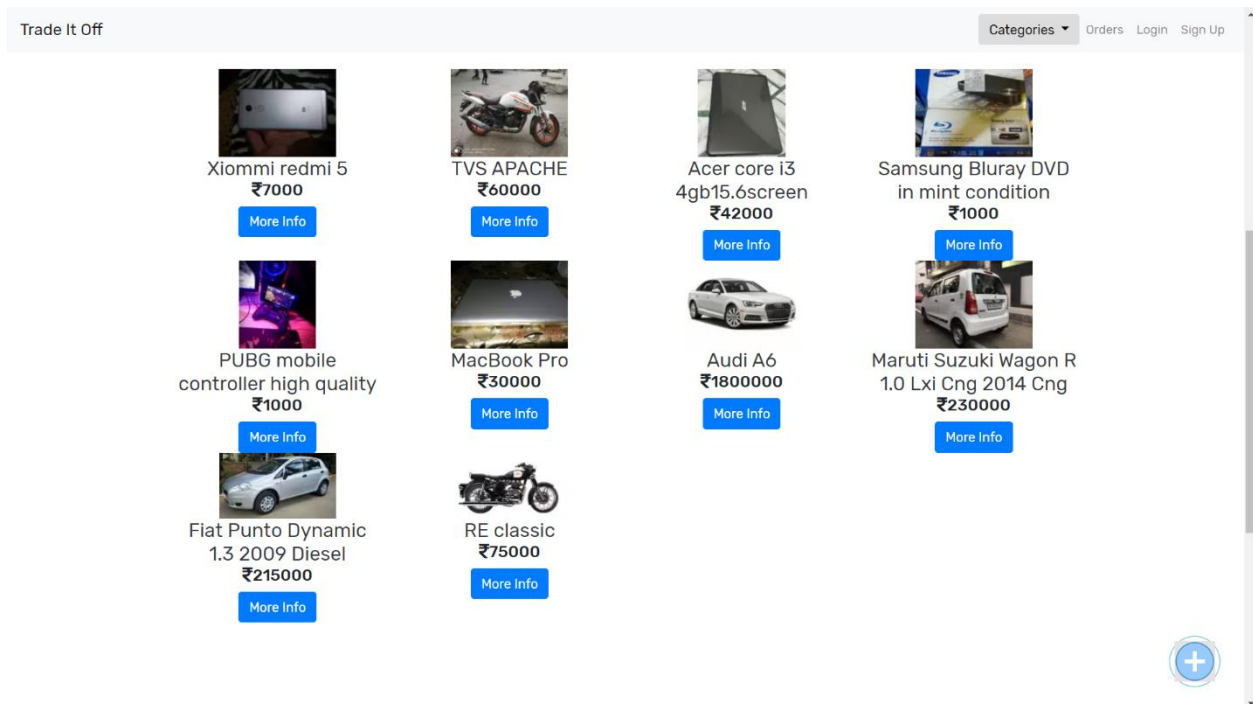


Fig 8.3 Products

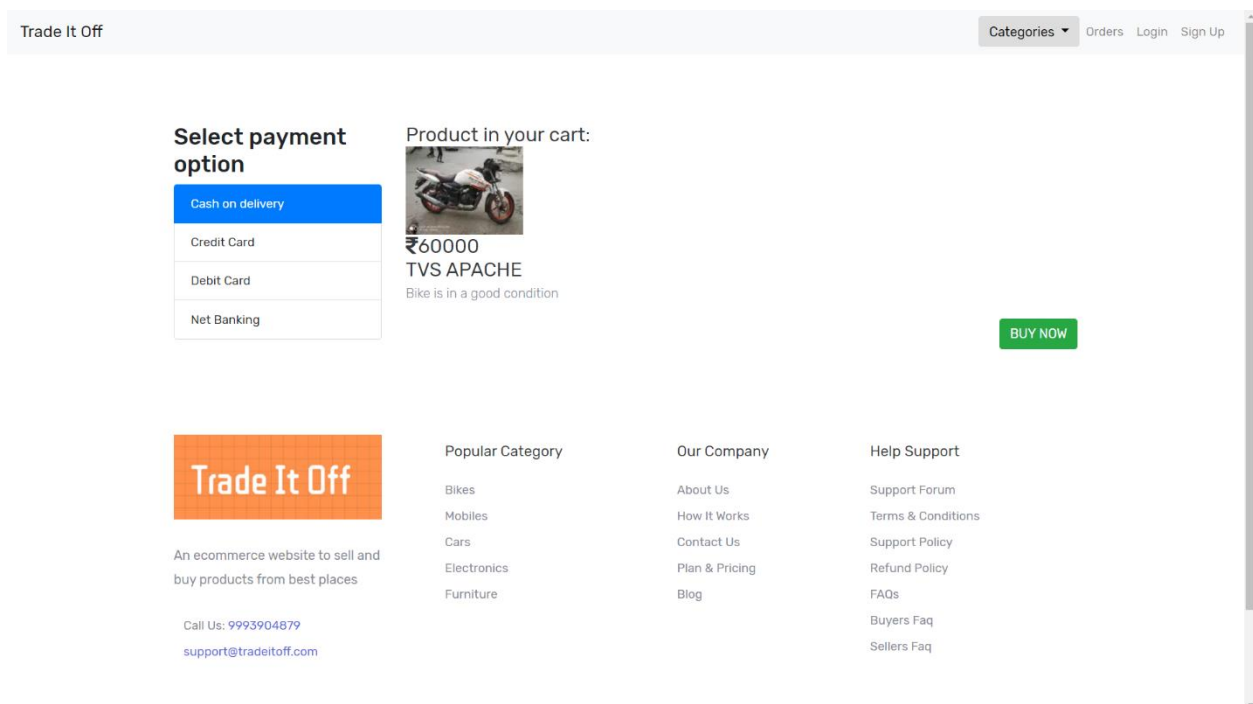


Fig 8.4 Product Details and Buy Now Page

CONCLUSION

The conclusion of this project is that, the project helps buyers and sellers to connect to each other and trade used as well as new products online at adequate prices. This platform will enhance the ecommerce trade over the internet.

In pursuit of achieving this target, a web application is more preferable as compared to a desktop application as it provides with real time updates and easy usage to a much wider population.

This has been a great experience for us, working as a team and learning together to expand our knowledge beyond curriculum per se. We would also like to express our gratitude towards our mentors who have been a guiding light throughout this journey.

REFERENCE

- **Express.js:** <https://www.tutorialspoint.com/expressjs/index.htm>
- **Node.js:** <https://www.w3schools.com/nodejs/>
- **Bootstrap:** <https://www.w3schools.com/bootstrap/>

