Chapter 1

**Introduction**

**1.1 Introduction**

**"BD – Tour For Local Guide"** is a Web based system providing all relevant and accurate information of tour guide’s job is to show people’s places and areas that they are not familiar with and give them information and facts about the place they are visiting.This system will provide a way to create connections among the tourists or travelers and places, so that they can travel more confidently and safely.

**1.2 Objective**

The main objective of this project is managed a tour guide’s job is to show people’s places and areas that they are not familiar with and give them information and facts about the place they are visiting. Which means the people or visitors will be able to know their choose able tourist places and also be able to hire a guide. We are going to make a system where people get facilities to hire guide for their choose able places & local guides can apply for guide job. Two major targets: one is creating employment facilities & another is secure guideline for travelers.

**1.3Benefits of the Project**

* Travelers can view the places by their choice.
* Here travelers can getsecure guide for reliable travel.
* Creating employment facilities.
* Secure guideline for travelers.
* Local people can apply for guide job.
* Travelers can pay online.

**1.4 Summary**

This chapter includes that, “BD - Tour For Local Guide”is most important tool for travelers and people. This chapter also describes the objective of our work where travelers can get secure guide for tour and local people can apply for guide job.

Chapter 2

**Background Study**

**2.1 Tourist Guide**

A tour guide or a tourist guide is a person who provides assistance, information on cultural, historical and contemporary heritage to people on organized sightseeing and individual clients at educational establishments, religious and historical sites such as; museums, and at various venues of tourist.  
**2.2 Tourist Guideconduct**

Tour Guides offer company and information to individuals or groups of travelers. They are knowledgeable and savvy about the places they show to tourists. On tours that include sightseeing historical buildings, sites, and locations, Tour Guides must be able to provide travelers with all the information they should know about the place they’re visiting. Most Tour Guides work on a pre-established route and visit a selected list of locations. They are usually employed by tour operators, hotels, resorts, and other establishments such as museums or historical landmarks.

**2.3 Existing System Overview**

There are no any proper existing system like as our local tourist guide system.

**2.4 Project Features**

Following are the features of our software:

* Tourist can be create profile.
* Tourist can be manage their bookings.
* Tourist can be book guide online.
* Tourist can be pay online.
* Tourist can be able to see order status online.

**2.5 Feasibility Study**

Feasibility study refers the viability of project in different terms. This stage ensures if the proposed project is viable in technical and business perspective. Usually feasibility study is an analysis step that determines if the proposed project is feasible in different aspects. This analysis stage helps to define the project feasibility if it is technically, operationally, socially, economically and ethically feasible.[7]

**2.6 Economic Feasibility**

This stage is undertaken to understand if the project is within budget and can make business benefits.

I think the proposed project is economically feasible. The reasons are given below,

As it can be used for tourist facilities and our tourist places will increase their acceptance to foreigners.[6]

**2.7 Technical Feasibility**

Technical feasibility refers if the identified project can be made efficiently with required functionality. It evaluates whole process how information will store and deliver on this project. Also evaluate if the organization is available to make this project.[5] Following matter will follow on this feasibility,

* No error is occurred on different functionality when occurred.
* As the project is platform independent, so it should run on different platform and on different device.

So it can be said, my project is technically feasible.[3]

**2.8 Operational Feasibility**

This term evaluate if the project solves all the functionality efficiently. How well it performs the functionality to avoid error and how efficiently it easily replaced the paper based system.

My system can easily utilize, support and perform all the functionality efficiently. All the stuff is done by the program. User doesn’t have to do anything.[7]

**2.9 Summary**

From this chapter we know about what a tourist guide is and also the conducting process of it. Here we also discuss about the different categories of facilities which should be included in a tourist guide.

.

Chapter 3

**Requirement Analysis**

**3.1 Requirement Analysis**

Requirement Analysis is the third step of System Development Life Cycle (SDLC). This step is very much important and inseparable part of a project. Requirement Analysis is concerned with discovering and deciding what the new system is required to do.  
**3.2 Process Specification**

* **Registration:** Users of this system must be registered firstly for using this system. User should give the name, Type (customer/vendor), email, Phone number, Password in the registration form. Without registration no-one access the backend.
* **Verification:** After giving the registration information a confirmation link will be send customer email, after customer confirms email customer account will be verified.
* **Login:** Users of this system must be login for using this system. User should give his email and password in the login form. System automatically checks the user’s information and gives the access the valid user into the system.
  1. **Problem Definition**

Manage shop is hard task for shop owner. If a shop owner likes to manage his/her shop and sell product online he had to create to software one is point of sell and other is ecommerce website. By our software shop owner will get both functionality.

* 1. **Developer’s Specification**

**3.4.1 Hardware specification**

* Ram: 4GB or above
* Processor: Intel Core i3 or above

**3.4.2 Software Specification**

* Operating System: Windows/ Linux
* Developed language: PHP
* Framework: Laravel
* Database: My-SQL
  1. **Platform**

Web Applications.

* 1. **My-SQL**

My-SQL is free and open-source software under the terms of the GNU General Public License, and is also available under a variety of proprietary licenses. My-SQL was owned and sponsored by the Swedish company My-SQL AB, which was bought by Sun Microsystems (now Oracle Corporation).[8] In 2010, when Oracle acquired Sun, Widenius forked the open-source My-SQL project to create MariaDB.[6]

* 1. **PHP**

PHP is one of the most admired and popular server side scripting languages which are widely used for creating websites. With faster turn-around time, enhanced security and affordability, PHP has become preferred choice of the website developers. Due to plethora of benefits of this scripting language, many famous online businesses such as Facebook, Yahoo, Wikipedia, Flickr etc. are using this scripting language. In this article, I am explaining 11 advantages that make PHP popular among web developers.

* 1. **Why use PHP?**
* It is one of the most popular programming language in the world
* It is easy to learn and simple to use
* It is open-source and free
* It is secure, fast and powerful
* It has a huge community support (tens of millions of developers)
  1. **Requirements Elicitation**

It is very important to grasp the scope of the system, what is the core functionality and what represents a nice to have' feature.

The elicitation step involves gathering clear and precise requirements, in order to model the system and its characteristics, a process that can be very complex in software development.

Due to the fact that this project is mainly focused on research and lesson providing a user oriented tool, it only uses the main techniques for analyzing the system's requirements.

**3.10 Functional Requirements**

In a nut shell, functional requirements describe what the software / website should do (the functions). These are:

* Login & verification process should work correctly.
* Different usability method for vendors and consumers must work properly.

Admin for managing the environment of this system needed to work fine as above**.**

**3.11 Non-functional Requirements**

Non-functional requirements define system attributes which is also known as system qualities such as performance, maintainability and scalability. These are:

* This system has to be available 24 hours a day.
* The system must be able to use several data formats according to the database.
* Admin must be able to maintain the system easily.

**3.12 Summary**

This chapter includes the projects the project’s functional and non-functional requirements to complete this system.

Chapter 4

**Proposed System Process Model**

**4.1 Proposed System Overview**

The following features should be added to a proposed system for tourist:

* View place online.
* Booking Guide online.
* Update profile information.
* Online payment.

The following features should be added to a proposed system for Guide:

* Request for job.
* Update profile.
* View his/her booking status.

**4.2 Agile Model**

**Agile development model is also a type of Incremental model. Software is developed in incremental, rapid cycles. This results in small incremental releases with each release building on previous functionality. Each release is thoroughly tested to ensure software quality is maintained. It is used for time critical applications.**

**4.3 Agile Manifesto principles**

* **Individuals and interactions** − In Agile development, self-organization and motivation are important, as are interactions like co-location and pair programming.
* **Working software** − Demo working software is considered the best means of communication with the customers to understand their requirements, instead of just depending on documentation.
* **Customer collaboration** − As the requirements cannot be gathered completely in the beginning of the project due to various factors, continuous customer interaction is very important to get proper product requirements.
* **Responding to change** − Agile Development is focused on quick responses to change and continuous development.

**4.4 Advantages of the Agile Model**

* It is a very realistic approach to software development.
* Promotes teamwork and cross training.
* Functionality can be developed rapidly and demonstrated.
* Resource requirements are minimum.
* Suitable for fixed or changing requirements
* Delivers early partial working solutions.
* Good model for environments that change steadily.
* Minimal rules, documentation easily employed.
* Enables concurrent development and delivery within an overall planned context.
* Little or no planning required.
* Easy to manage.
* Gives flexibility to developers.

## 4.5 Methodology of Agile Model

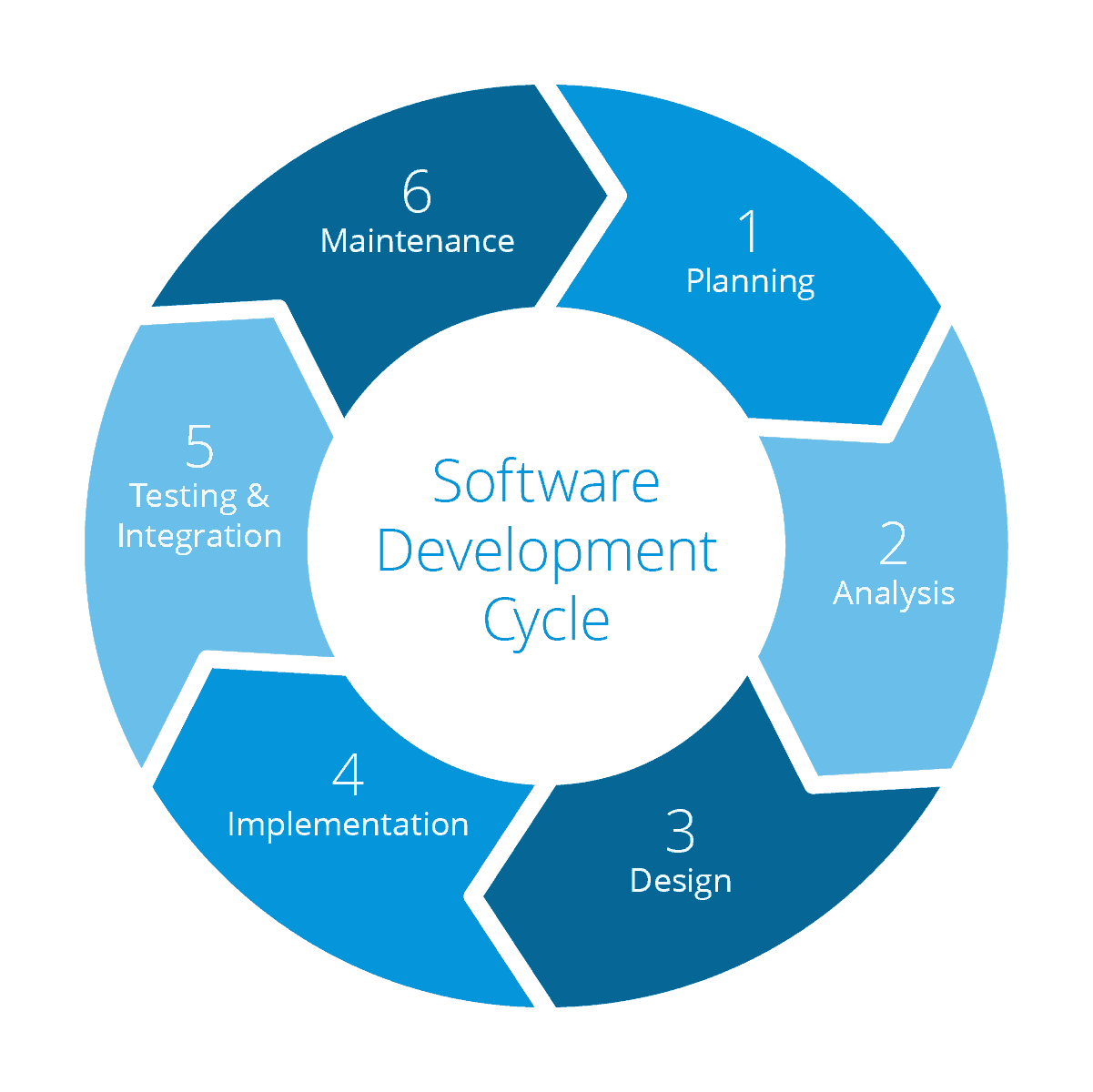
AGILE methodology is a practice that promotes **continuous iteration** of development and testing throughout the software development lifecycle of the project. Both development and testing activities are concurrent unlike the Waterfall model.

The agile software development emphasizes on four core values.

* Individual and team interactions over processes and tools
* Working software over comprehensive documentation
* Customer collaboration over contract negotiation
* Responding to change over following a plan

**4.6 Workflow in Agile Model**

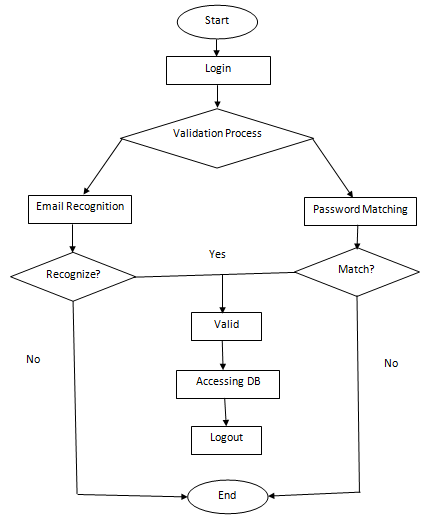
* **Planning:** Captured in a product requirement document.
* **Analysis:** Resulting in models, schema and bounding rules.
* **Design:** Resulting in the software architecture.
* **Implementation:** The development, providing and integration of software.
* **Testing& Integration:** The systematic discovery and debugging of defects.
* **Maintenance:** the installation, migration, support and maintenance of complete system.



**Figure 4.2: Agile Model**

**4.7 Proposed System Flowchart**

Here is our system flowchart, which shows how our system will work.

**  
Figure 4.2: Proposed System Flowchart**

**4.8 Summary**

This chapter includes the process model that has followed. Waterfall model is followed to complete this project which is very simple to work with. This chapter also shows how the system will work in the figure.

Chapter 5

**System Design**

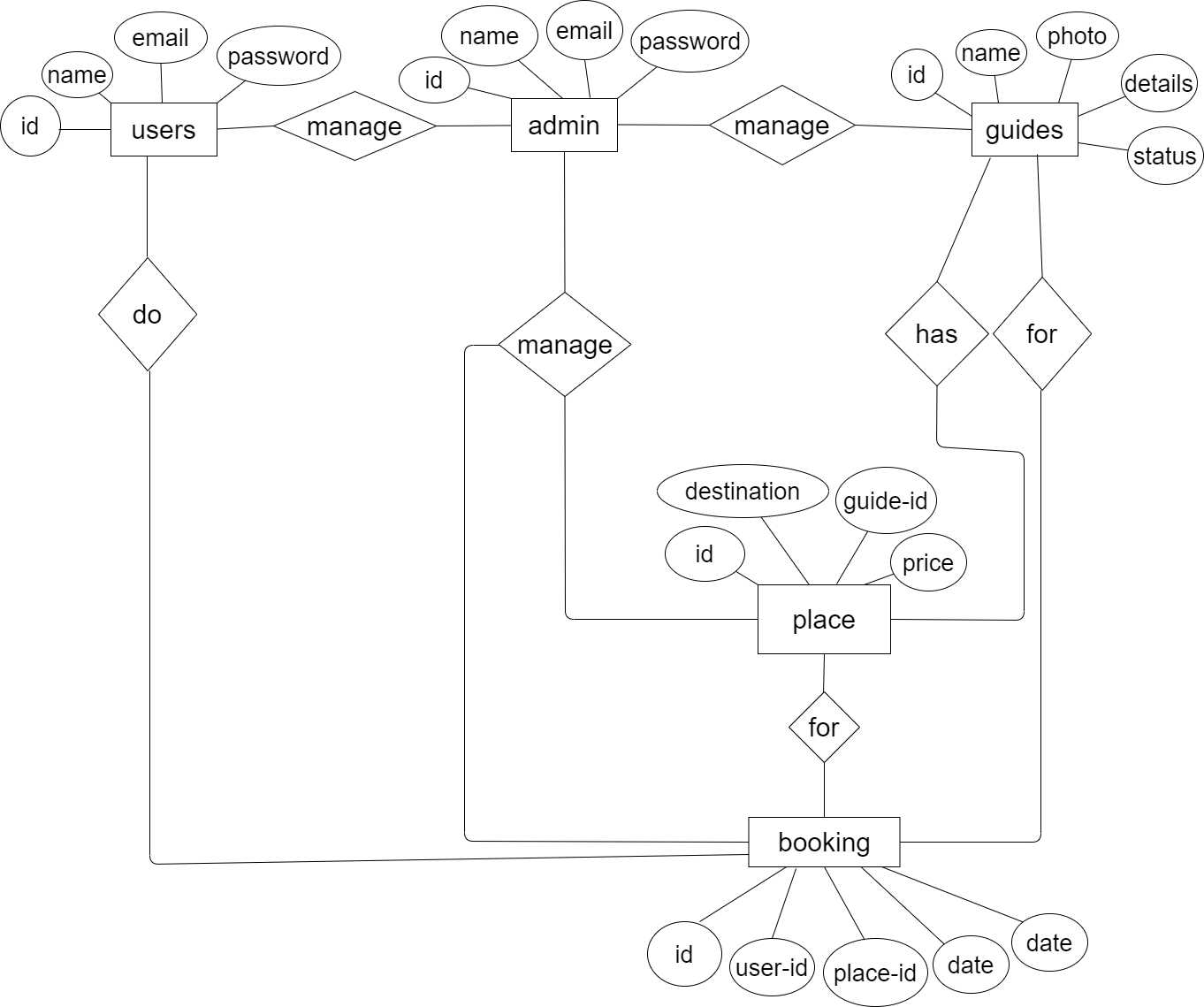
**5.1 E-R Diagram**

In software engineering, an Entity-Relationship Model is an abstract and conceptual representation of data. Entity-relationship modelling is a database modelling method, used to produce a type of conceptual schema or semantic data model of a system, often a relational database, and its requirements in a top-down fashion.

There are three types of relationships that can exist between two entities.   
An entity-relationship (ER) diagram is created based on these three types, which are listed below:

* **one-to-one relationship**: In relational database design, a one-to-one (1:1) relationship exists when zero or one instance of entity A can be associated with zero or one instance of entity B, and zero or one instance of entity B can be associated with zero or one instance of entity A. (abbreviated 1:1)
* **one-to-many relationship**: (abbreviated 1:N) In relational database design, a one-to-many (1:N) relationship exists when, for one instance of entity A, there exists zero, one, or many instances of entity B; but for one instance of entity B, there exists zero or one instance of entity A.
* **many-to-many relationship**: In relational database design, a many-to-many (M:N) relationship exists when, for one instance of entity A, there exists zero, one, or many instances of entity B; and for one instance of entity B, there exists zero, one, or many instances of entity A.

Diagrams created using this process are called entity-relationship diagram or E-R Diagram or ERD.



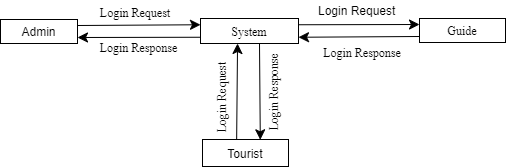
**Figure 5.1: E-R Diagram**

**5.2 Data Flow Diagram**

A data flow diagram is a graphical depiction of flow of data through intended software system and is used asthe1st step to creating an overview of the system. It’s really useful as it provides an overview of data as well as functionally to software designers.

**5.2.1 DFD Level 0**

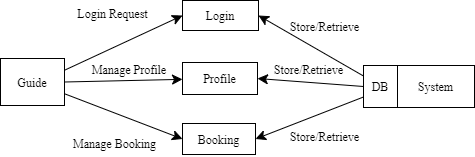
Here we are showing the high level process of the system. It’s a basic overview of the whole multivendor e-trading:



**Figure 5.2.1:** 0-Level DFD

**5.2.2 DFD Level 1**

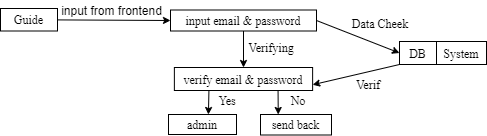
The First level DFD of this system shows how the system is divided into sub-systems (processes), each of which deals with one or more of data flows to or from an external agent.



**Figure 5.2.2:** 1-Level DFD

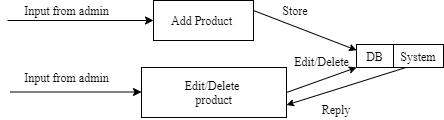
**5.2.3 DFD Level 2**

The Second level DFD goes one step deeper into parts of Level 1 of the system, where we have elaborated the low level process of the system.



**Figure:**DFD Label 2 Process 1

**5.2.3.1 Level 2 Process 2**



**Figure:** DFD Label 2 Process 2

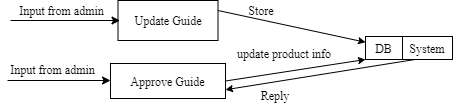
**5.2.3.2 Level 2 Process 3**

**Figure:** DFD Label 2 Process 3

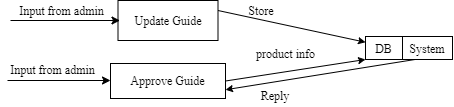


**5.2.3.3 Level 2 Process 4**

**Figure:** DFD Label 2 Process 4



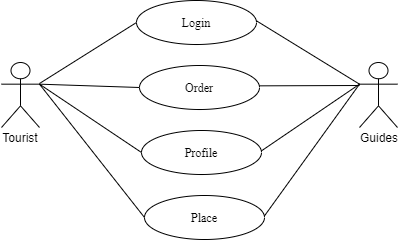
**5.2.3.4 Level 2 Process 5**



**Figure:** DFD Label 2 Process 5

**5.3 Use Case Diagram**

This figure shows that how the system can be used by the Users and Admin panel. Obviously the using capability of an Admin is not same as a user.



**Figure 5.5: Use Case Diagram**

**5.4 Summary**

This chapter shows E-R diagram for showing the entity relation in our database. This chapter discusses data flow diagram for showing the flow of data such as DFD level 0, DFD level 1 and DFD level 2. It also includes use case diagram of the system.

Chapter 6

**Implementation**

* 1. **Technology Used In Application**

|  |  |
| --- | --- |
| **Front-end** | **Back-end** |
| HTML5, CCS3, Bootstarp4 | PHP7 , My-SQL, Laravel |

* 1. **Hardware / Software Requirement**

Minimum software requirement and hardware requirement to use this application is given below.

* + 1. **Minimum Software Requirement**

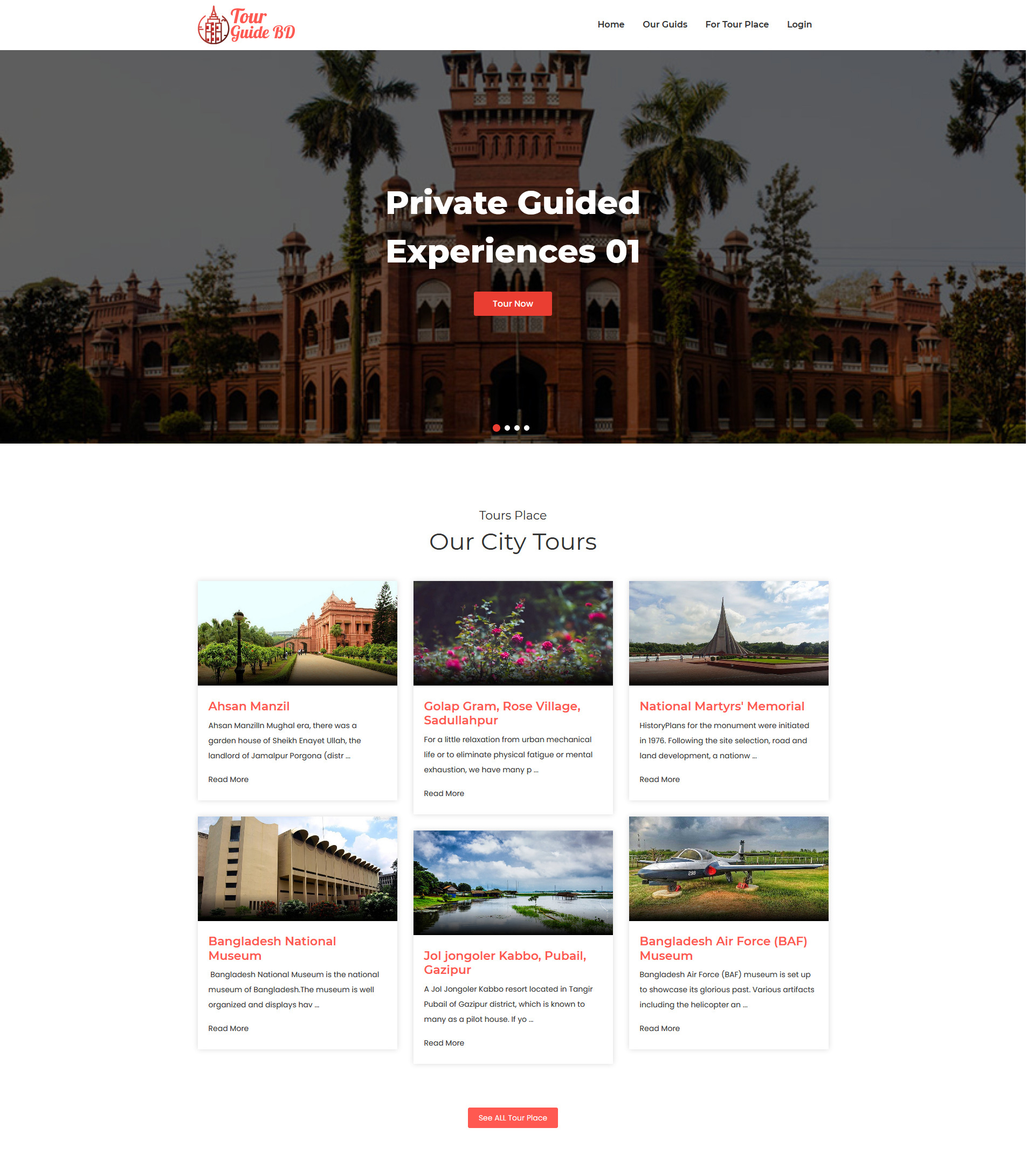
|  |  |
| --- | --- |
| Operating System | Windows xp or Any compatible OS |
| Web Browser | IE 6.0 or Any compatible web browser |
| XAMPP | XAMPP-VM / **PHP** 7.2.9 |

* + 1. **Minimum Hardware Requirement**

|  |  |
| --- | --- |
| Processor | 1 MHZ |
| Ram | 512 |
| Hard Disk | 10GB (free space) |

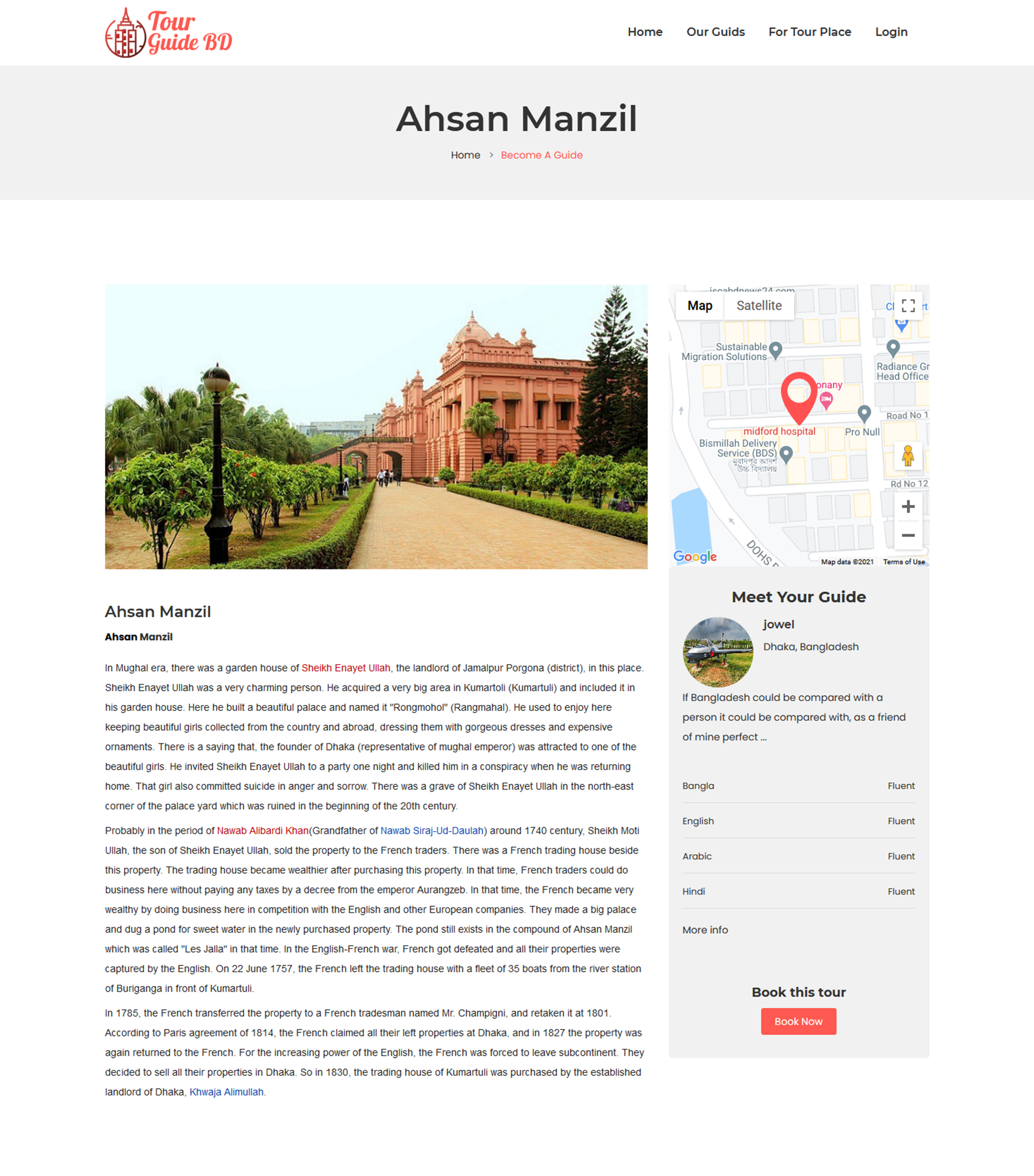
**6.3Project Showcasing**

**6.3.1 Home Page**



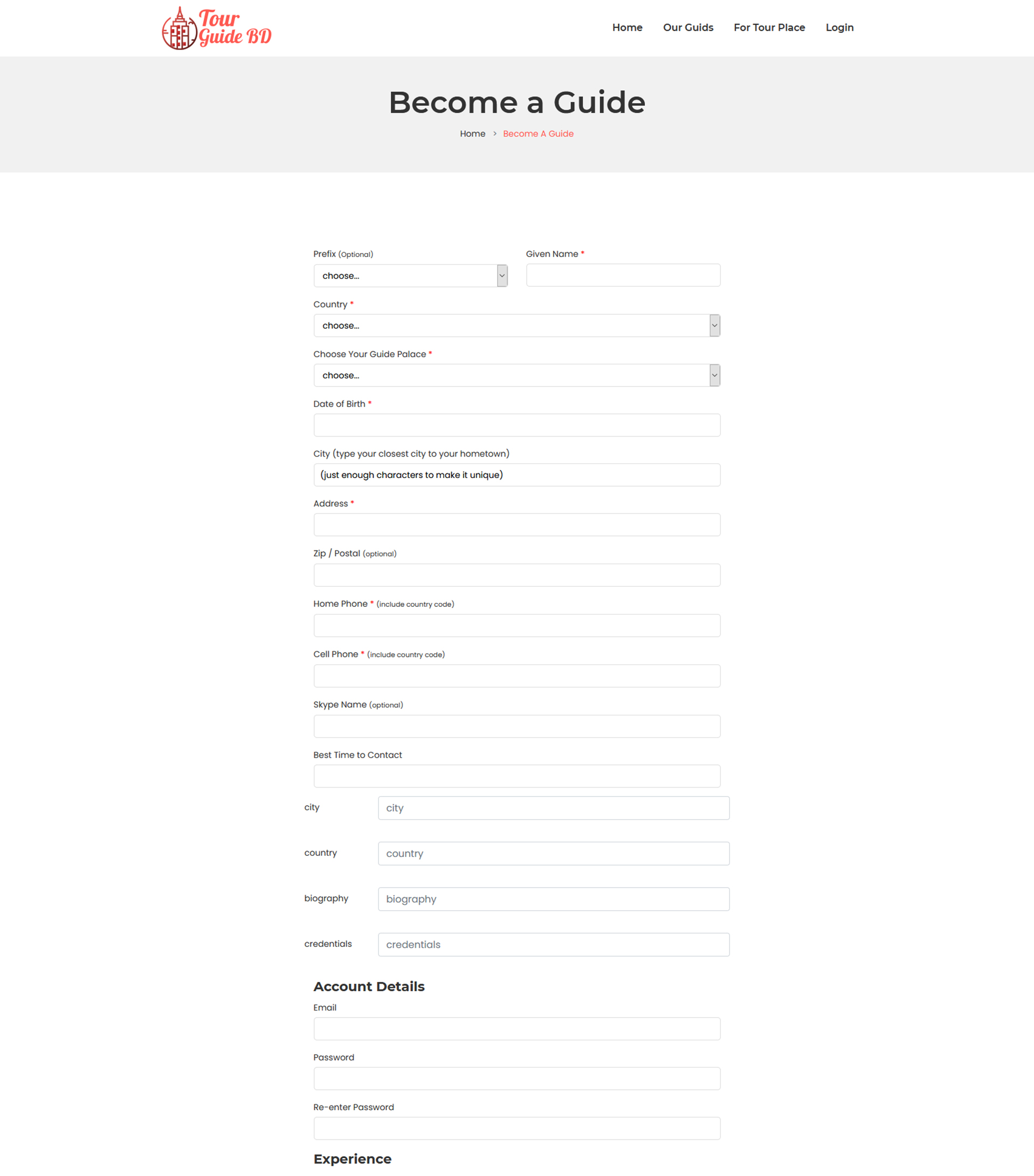
**Figure6.3.1:**Home Page

**6.3.2 Tour Details Page**



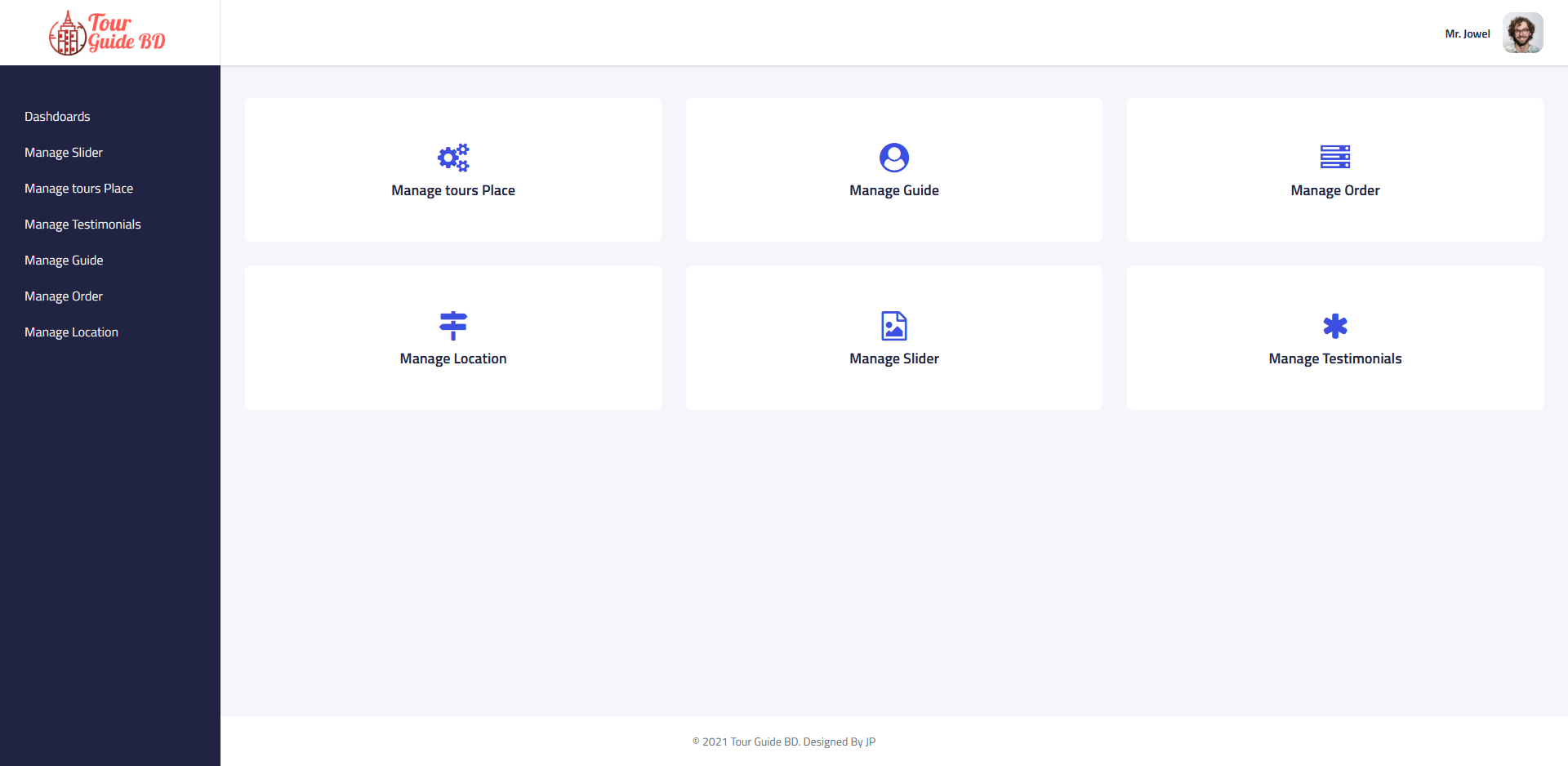
**Figure6.3.2:**Tour Details Page

**6.3.3 Guide Registration**



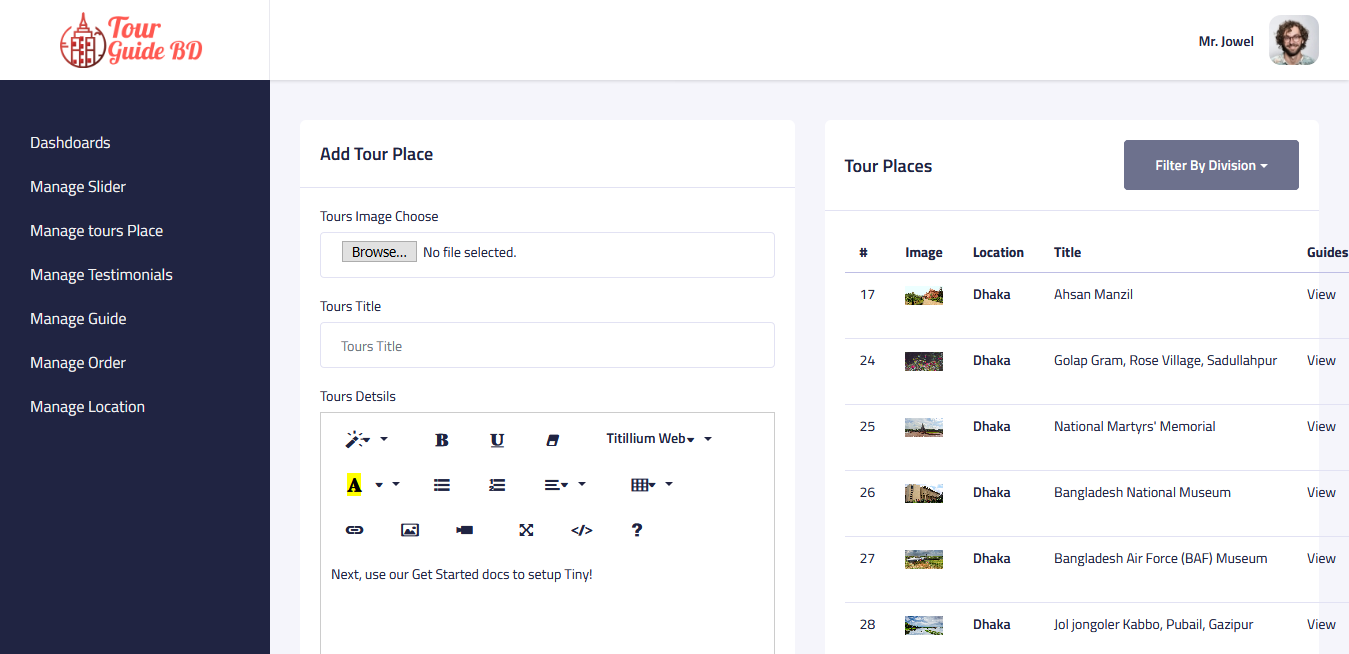
**Figure6.3.3:**Guide Registration

**6.3.4 Super Admin Dashboard**



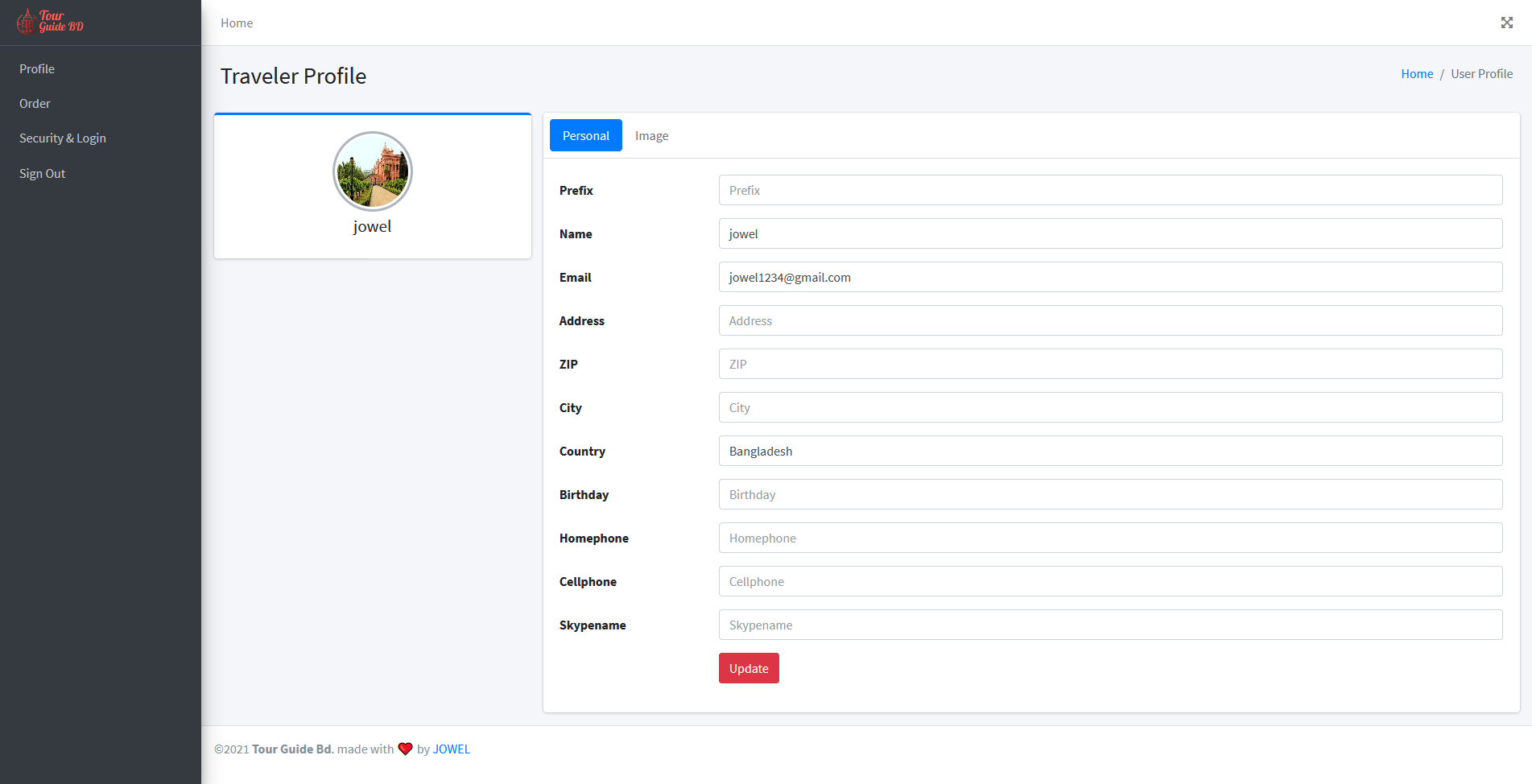
**Figure6.3.4:**Super Admin Dashboard

**6.3.5Super Admin Tour Place Manage**



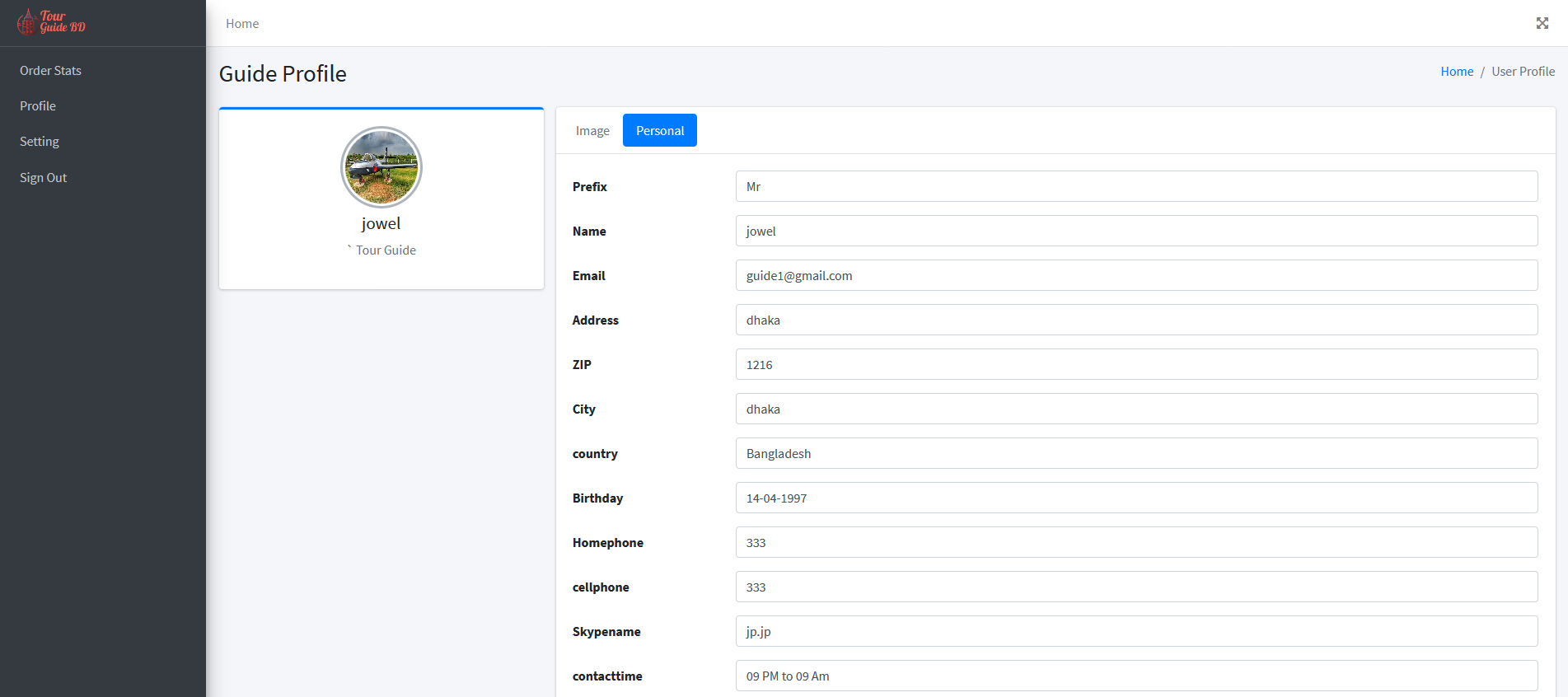
**Figure 6.3.5:** Super Admin Tour Place Manage

**6.3.6 Traveler Profile**



**Figure 6.3.6:** Traveler Profile

**6.3.4 Guide Profile**



**Figure 6.3.7: Guide Profile**

Chapter 7

**Testing**

**7.1 Testing**

Software testing is an activity to check whether the actual results match the expected results and to ensure that the software system is defect free. It involves execution of a software component or system component to evaluate one or more properties of interest.

Software testing also helps to identify errors, gaps or missing requirements in contrary to the actual requirements. It can be either done manually or using automated tools. Testing can be both manual ans also automation.

**7.2 Types of Testing**

Typically Testing is classified into three categories.

* Functional Testing
* Non-Functional Testing
* Maintenance

**7.2 .1 Functional Testing**

Functional Testing is a quality assurance (QA) process and type of black box testing that based on its test cases on the specification of the software component under test cases .

Different types of functional testing are given below:

* Unit Testing
* Integration Testing
* Smoke
* UAT (User Acceptance Testing)
* Localization
* Globalization
* Interoperability

**7.2.2 Non-Functional Testing**

Non-Functional Testing is a testing of a software application or system for its non-Functional requirements the way a system operates , rather than specific behaviors’ of the system

Different types of non-Functional testing are given below:

* Performance
* Endurance
* Load
* Volume
* Scalability
* Usability

**7.2.3 Maintenance**

Once the system is deployed and it is in service for yours and decades. During this time the system and its operational environment is often corrected, changed or extended.

Different types of non-Functional testing are given below:

* Regression
* Maintenance

**7.3 Unit Testing**

Unit testing is a level of of software testing where individual units/components are tested. The purpose is to validate that each unit of the software performs are designed

**7.4 System Testing**

System Testing is a level of the software testing where complete and integrated software is tested. The purpose of this test is to evaluate the system’s compliance with specified requirements.

**7.5 Test Cases of the System**

**Table: 7.5.1 Test Cases**

|  |  |  |
| --- | --- | --- |
| **SL NO** | **Test Objectives** | **Test Process** |
| 1 | Check Database Connection | <http://localhost/phpmyadmin>  Create a database in this server  Enter database username  Enter database password  Enter database name |
| 2 | Check Guide Registration Form | <http://127.0.0.1:8000/guide-apply>  Enter a unique email address  Enter password  Choose prefix  Choose name  Choose country  Choose guide place  Choose date of birth  Choose city  Enter address  Click for submit |
| 3 | Check Traveler Registration From | <http://127.0.0.1:8000/traveler-register>  Enter a unique email address  Enter password  Choose full name  Choose country  Click for submit |
| 4 | Check Login Credentials | <http://127.0.0.1:8000/login>  Enter email address  Enter password |

**Table: 7.5.2 Check Database Connections**

|  |  |  |  |
| --- | --- | --- | --- |
| **TEST CASES** | **TEST DATA** | **EXPECTED RESULT** | **ACTUAL RESULT** |
| Valid server name, username, password, database name  Valid server name, invalid username, password, valid database name  Invalid server name, valid username, password, database name  Valid server name, username, password, invalid database name | Localhost, root, NULL, QB  Localhost, boot, NULL, QB  Publichost, root, NULL, QB  Localhost, root, NULL, BB | Successful database connection  Unsuccessful database connection  Unsuccessful database connection  Unsuccessful database connection | Successful  Failed  Failed  Failed |

**Table: 7.5.3Check Guide Registration**

|  |  |  |  |
| --- | --- | --- | --- |
| **TEST CASES** | **TEST DATA** | **EXPECTED RESULT** | **ACTUAL RESULT** |
| Valid Gmail, password, name, country  Invalid Gmail, valid password, name, country  Valid (duplicate Gmail), password, name, country | [jowelmurad@gmail.com](mailto:mazharul@gmail.com), 12345, Jowel Murad, Bangladesh  [jowelmurad.com](mailto:mazharul@gmail.com), 12345, Jowel Murad,Bangladesh  [jowelmurad@gmail.com](mailto:jowelmurad@gmail.com), 55555, Jowel Murad, Bangladesh | Registration Successful  Registration Unsuccessful  Gmail already taken, please login | Successful  Failed  Failed |

**Table: 7.5.4 Check Traveler Registrations From:**

|  |  |  |  |
| --- | --- | --- | --- |
| **TEST CASES** | **TEST DATA** | **EXPECTED RESULT** | **ACTUAL RESULT** |
| Valid Gmail, password, name, country  Invalid Gmail, valid password, name, country  Valid (duplicate Gmail), password, name, country | [jowelmurad@gmail.com](mailto:jowelmurad@gmail.com), 12345, Jowel Murad, Bangladesh  [jowelmurad.com](mailto:mazharul@gmail.com), 12345, Jowel Murad, Bangladesh  [jowelmurad@gmail.com](mailto:jowelmurad@gmail.com), 55555, Jowel Murad, Bangladesh | Registration Successful  Registration Unsuccessful  Gmail already taken, please login | Successful  Failed  Failed |

**Table: 7.5.4 Check Login Credentials:**

|  |  |  |  |
| --- | --- | --- | --- |
| **TEST CASES** | **TEST DATA** | **EXPECTED RESULT** | **ACTUAL RESULT** |
| Correct email & password  Correct email & incorrect password  Incorrect email & correct password  Incorrect email & password | [jowelmurad@gmail.com](mailto:jowelmurad@gmail.com), 12345  [jowelmurad@gmail.com](mailto:jowelmurad@gmail.com), tttttt$$#%  [jowelmurad@gmail.com](mailto:jowelmurad@gmail.com), 12345  [jowelmurad@gmail.com](mailto:jowelmurad@gmail.com), tttttt$$#% | Login Successful  Wrong password  Invalid email  Invalid email | Successful  Failed  Failed  Failed |

**7.6 Benefits of Testing**

* Maintains the quality of the software
* Make sure that system meet all its requirements
* Makes the software as per client’s requirement
* Unit testing makes each module more perfect of the system
* System testing makes the entire application defect free.

**7.7 Summary**

This chapter describes briefly about software testing. It has a description about different types of testing and also discussion about the unit and system testing of our system. At the end of this chapter includes the benefits of testing in terms of maintaining the quality of a product.

Chapter 8

**Future Work**

**&**

**Conclusion**

**8.1 Future Work**

In future we have a plan to implement Artificial Intelligence in this System. As it is a web based application now we also have a plan to develop Mobile App on current system.

**8.2Conclusion**

In this “Tour Guide BD” system is managed a tour guide’s job is to show people’s places and areas that they are not familiar with and give them information and facts about the place they are visiting. Which means the people or visitors will be able to know their choose able tourist places and also be able to hire a guide. We are going to make a system where people get facilities to hire guide for their choose able places & local guides can apply for guide job. Two major targets: one is creating employment facilities & another is secure guideline for travelers.

**References**

[1] Software Engineering: A Practitioner's Approach Book by Roger S. Pressman, March 12, 1982

[2] Software engineering for student by Douglas Bell. -- 4th ed. p. cm , Rev. ed. of: Software engineering. 2000

[3] Software Engineering: A Practitioner's Approach by Roger S. Pressman. – 7th ed.

[4] Melao, N. (2008), “E-Business Processes and E-Business Process Modeling: A State-of-the-Art Overview”, pp. 54-89.  
  
[5] Mendo, F. and Fitzgerald, G. (2005), “Theoretical Approaches to Study SMEs E-Business Progression”, Journal of Computing and Information Technology, Vol. 13, No. 02, pp. 123-136.  
  
[6] Poon S. & Swatam (1997), “Small business use of Internet: Findings from Australian case studies”, Internation Marketing Review. Pp. 385-403.

[7] Rahmath S., Hema D. and Abdullah K. (2011). “International Arab Journal of e-Technology”, Vol. 2, No. 1, pp.56-64.

[8] Ramboll Management ‘ICT and E-Business Impact in the Banking Industry’, Version 4.0,

**APPENDIX**

**Source Code:  
Home Page:**

 <!-- slider-main-area start -->  
    <div class="slider-main-area">  
        <div class="slider-active owl-carousel">  
            <!-- slider-wrapper start -->  
            @foreach($sliders as $slider)  
            <div class="slider-wrapper bg-overlay" style="background-image:url({{ asset('assets/images/gallery/'.$slider->image) }})">  
                <div class="container">  
                    <div class="row">  
                        <div class="col-lg-8 ml-auto mr-auto">  
                            <div class="hero-text-content">  
                                <h1 class="text-white">{{$slider->title}} </h1>  
                                <div class="slider-button-box">  
                                    <a title="Tour now" href="{{route('front.fortour')}}" class="slier-btn">Tour Now</a>  
                                </div>  
                            </div>  
                        </div>  
                    </div>  
                </div>  
            </div>  
            @endforeach  
            <!-- slider-wrapper end -->  
        </div>  
    </div>  
    <!-- slider-main-area end -->  
  <!-- Tour Place Area Start -->  
    <div class="tour-place-area masonry-activation section-ptb">  
        <div class="container">  
  
            <div class="row">  
                <div class="col-lg-12">  
                    <div class="section-title text-center mb-5">  
                        <h4>Tours Place</h4>  
                        <h3>Our City Tours</h3>  
                    </div>  
                </div>  
            </div>  
  
            <div class="row clearfix masonry-wrap">  
@foreach($tours as $tour)  
                <div class="col-lg-4 col-md-6 masonary-item">  
                    <article class="single-tour-place">  
                        <div class="image-thumb">  
                            <a class="featured\_image" href="./single-tour-details/{{$tour->id}}" title="{{ $tour->title }}">  
                              <img src="{{asset('assets/images/tour-place/'.$tour->image)}}" alt="{{ $tour->title }}">  
                            </a>  
                            <div class="location">  
                                <!-- <a href="{{$tour->map\_link}}"><i class="fa fa-map-marker"></i></a><a href="/single-tour-details/{{$tour->id}}" rel="category tag">{{ $tour->title }}</a> -->  
                                <!-- <a href="{{$tour->map\_link}}" rel="category tag"><i class="fa fa-map-marker"></i> Map Link</a> -->  
                            </div>  
                        </div>  
                        <div class="tour-small-info">  
                             
                            <h4 class="title-post"><a href="./single-tour-details/{{$tour->id}}" rel="bookmark">{{ $tour->title }}</a></h4>  
                             
                            <p>  
                                {{ substr(strip\_tags($tour->details), 0, 120) }}  
                                {{ strlen(strip\_tags($tour->details)) > 120 ? "..." : "" }}</p>  
                            <a class="read-more" href="./single-tour-details/{{$tour->id}}">{{ strlen(strip\_tags($tour->details)) > 120 ? "Read More" : "" }}</a>  
                        </div>  
                    </article>  
                </div>  
                 @endforeach  
               </div>  
<div class="row">  
                <div class="col-lg-12">  
                    <div class="all-item-show-button-box text-center mt-5">  
                        <a href="{{route('front.fortour')}}" class="button">  
                            See ALL Tour Place  
                        </a>  
                    </div>  
                </div>  
            </div>  
</div>  
    </div>  
    <!-- Tour Place Area End  -->

**Admin Page:**

       <!-- Content Body Start -->  
        <div class="content-body">  
  
            <div class="manage-slier-area">  
  
  @if ($message = Session::get('success'))  
  
                    <div class="alert alert-success alert-block mt-3">  
  
                        <button type="button" class="close" data-dismiss="alert">×</button>  
  
                        <strong>{{ $message }}</strong>

</div>  
@endif  
  
                  @if (count($errors) > 0)  
  
                        <ul class="alert alert-danger pl-5">  
  
                          @foreach($errors->all() as $error)  
  
                             <li>{{ $error }}</li>  
  
                          @endforeach  
  
                        </ul>  
  
                @endif

**Guide Login:**

   <!-- Breadcrumb -->  
    <div class="breadcrumb-area">  
        <div class="container">  
            <div class="in-breadcrumb">  
                <div class="row">  
                    <div class="col">  
                        <h3>Tour Guide Login</h3>  
                        <!-- breadcrumb-list start -->  
                        <ul class="breadcrumb-list">  
                            <li class="breadcrumb-item"><a href="{{route('front.index')}}">Home</a></li>  
                            <li class="breadcrumb-item active">Become a Guide</li>  
                        </ul>  
                        <!-- breadcrumb-list end -->  
                    </div>  
                </div>  
            </div>  
        </div>  
    </div>  
    <!--// Breadcrumb -->  
     
     
    <div class="tourist-guide-apply-area section-ptb">  
        <div class="container">  
            <div class="row">  
                <div class="col-lg-7 ml-auto mr-auto">  
                    <!-- <p> Travel Agents, please use the <a class="theme-color" href="{{route('front.guidelogin')}}"> TRAVEL AGENT login</a></p> -->  
                    <p> Travelers, please use the  <a class="theme-color" href="{{route('front.travelerlogin')}}"> Traveler login.</a></p>  
                    <form name="theForm" method="post" class="login-form background-form" action="{{ route('login') }}">  
                    @csrf  
                        <div class="form-section">  
                            <div class="form-field">  
                                <label>Email</label>  
                                <div class="input-holder">  
                                    <input placeholder="e.g., [youremail@email.com](mailto:youremail@email.com)" tabindex="1"  id="email" type="email" name="email" >  
                                </div>  
                            </div>  
                            <div class="form-field">  
                                <label>Password</label>  
                                <div class="input-holder">  
                                    <input type="password" tabindex="2" id="password" name="password">  
                                </div>  
                            </div>  
                        </div>  
                        <input type="checkbox" name="31.11" value="31.11">&nbsp;Remember login on this computer  
                        <input type="hidden" name="wosid" value="#">  
                        <div class="text-center">  
                            <input tabindex="3" class="button green" type="submit" value="Login" name="31.7">  
                        </div>  
                        <br>  
                        <!-- <a href="{{ route('password.request') }}">forgot password?</a> -->  
                             
                    </form>  
                    <p class="join-text">If you are a passionate and enthusiastic person with local knowledge and expertise, <a href="{{route('guide.apply')}}" class="theme-color">join&nbsp;our&nbsp;system</a> and become a Tours By Locals tour guide!</p>  
                </div>  
            </div>  
        </div>  
    </div>