**Natours**

# A Thesis submitted to

**Chhattisgarh Swami Vivekanand Technical University Bhilai (C.G.), India**

*In partial fulfillment*

***For the award of the Degree***

***of***

# Bachelor Of Technology

# in

# Computer Science and Engineering

# by

# Devesh Gautam

# C Krishnaveni

# Abhijeet Singh

# B V Himanshu Rao

**Enrollment No: BH3865, BH3858, BH3813, BH3857**

**University Roll No.: 303602219057, 303602219050, 303602219005,** **303602219049**

Under the Guidance of

## R.K Khare

## Professor

**Computer Science and Engineering**

**Shri Shankaracharya Technical Campus**

**Junwani, Bhilai, Chhattisgarh, 490020**

**Session: 2019 - 23**

# Declaration by the Candidate

I the undersigned solemnly declare that the report of the thesis work entitled **“*Natours”***, is based on my own work carried out during the course of my study under the supervision of ***R.K Khare.***

I assert that the statements made, and conclusions drawn are an outcome of the project work. I further declare that to the best of my knowledge and belief that the report does not contain any part of any work which has been submitted for the award of any other degree/diploma/certificate in this University/deemed University of India or any other country. All helps received and citations used for the preparation of the thesis have been duly acknowledged.

(Signature of the Candidate)

Name of the Candidate- Devesh Gautam

Roll No.:303602219057

Enrollment No.:BH3865

**Signature of the Supervisor**

**R.K Khare**

**Professor**

**Computer Science and Engineering**

**Shri Shankaracharya Technical Campus**

**Junwani, Bhilai, Chhattisgarh, 490020**

# Certificate of the Supervisor

This is to certify that the report of the thesis entitled “***Natours”***, is a record of bonafide research work carried out by ***Devesh Gautam, C.Krishnaveni, Abhijeet Singh, B.V Himanshu Rao*** bearing Roll No- 303602219057, 303602219005, 303602219049 & Enrollment No.- BH3865, BH3858**,** BH3813, BH3857 under my guidance and supervision for the award of Degree of Bachelor of Engineering in the faculty of ***Computer Science and Engineering***, of Chhattisgarh Swami Vivekanand Technical University, Bhilai (C.G.), India. To the best of my knowledge and belief the thesis

* Embodies the work of the candidate him/herself,
* Has duly been completed,
* Fulfils the requirement of the ordinance relating to the BE/B.Tech degree of the University and is up to the desired standard both in respect of contents and language for being referred to the examiners.

(Signature of the Candidate)

Designation: Department:

Name:

Name & address of the Institute

Forwarded to Chhattisgarh Swami Vivekanand Technical University, Bhilai

(Signature of the Director/Principal)

Shri Shankaracharya Technical Campus

Junwani, Bhilai, Chhattisgarh,490020

# Certificate by the Examiners

The Thesis entitled “***Natours”*** Submitted by ***Devesh Gautam, C.Krishnaveni, Abhijeet Singh, B.V Himanshu Rao*** (Roll No.: 303602219057, 303602219077, 303602219107 Enrollment No.- BH3865, BH3885**,** BH3915) has been examined by the undersigned as a part of the examination and is hereby recommended for the award of the degree of Bachelor of Technology in the faculty of ***Computer Science and Engineering*** of Chhattisgarh Swami Vivekanand Technical University, Bhilai.

Internal Examiner External Examiner

Date: Date:

# Abstract

In today’s fast-changing business environment, it’s extremely important to be able to respond to client needs in the most effective and timely manner. If your customers wish to see your business online and have instant access to your products or services.

We made a complete generic template web-App which can be modified and manipulated to become any kind of supervision web-app for any kind of business or regular requirements with little to no efforts.

Supervision or Management website have been very common type of projects from generations now so we have tried to make improvements by making it generic and adding search features with very easy to use user experience and high-end authentication and we have used the most main stream technology to implement our idea.

# Acknowledgment

Working for the project has been a great experience for us. There were moments of anxiety, when we could not solve a problem for several days. But we have enjoyed the process and are thankful to all people associated with us during this period.

We convey our sincere thanks to our project guide **Prof. R.K Khare,** for providing us all sorts of facilities. Her support and guidance helped us to carry out the project. We are grateful for her constant advice, support, cooperation & encouragement throughout the project.

We would like to express the deep gratitude to respected **Prof. R.K Khare,** for his ever helping and support. We also pay special thanks for helpful solutions and comments enriched by his experience, which improved our ideas for betterment of the project.

We would also like to express our deep gratitude to our college management **Shri**

**I.P. Mishra, Chairman (Shri Gangajali Education Society, Bhilai), Mrs. Jaya Abhishek Mishra, President (Shri Gangajali Education Society, Bhilai), Dr.**

**P.B. Deshmukh Director (SSTC) & Dr. Siddharth Chaubey (HoD)** for providing an educational ambience. It will be our pleasure to acknowledge. Utmost cooperation and valuable suggestions from time to time given by our staff members of our department to whom we owe our entire computer knowledge and also we would like to thank all those persons who have directly or indirectly helped us by providing books and computer peripherals and other necessary amenities which helped us in the development of this project which would otherwise have not been possible.

(Signature of the student)

Name:

Shri Shankaracharya Technical Campus

Junwani, Bhilai, Chhattisgarh, 490020

# Table of Contents

[Declaration by the Candidate i](#_TOC_250006)

[Certificate of the Supervisor ii](#_TOC_250005)

[Certificate by the Examiners iii](#_TOC_250004)

Abstract iv

[Acknowledgment v](#_TOC_250003)

Table Of Contents vi

List of Tables vii

List of Figures viii

List of Symbols ix

List of Abbreviations x

Chapter-1 Introduction 11

Chapter – 2 Literature Review or Background Information: 13

Chapter -3 Methodology or Materials and Methods………….......................................……...15

3.1 Introduction..........................................................................................................15

3.2 Justification for the Methodology........................................................................16

3.3 System Requirements And Analysis..............................................……………..18

3.3.1 System analysis......................................................................................18

3.3.2 System Feasibility..................................................................................19

3.4 Requirement Analysis & Specification................................................................20

3.4.1 Requirements Gathering........................................................................21

3.4.2 Hardware Requirements....................................................................…22

3.4.3 Software Requirements....................................................................…22

3.5 System Design.....................................................................................................24

3.5.1 Design Goals...........................................................…...........................24

3.5.2 MVC Architecture.................................................................................25

3.5.3 Data Model Diagram.......................................................…..................26

3.5.4 JWT Authentication..............................................................................27

3.5.5Stripe Workflow .............................................................................................28

3.6 Architectural Design.............................................................................................29

3.6.1Architectural Context Diagram...............................................................29

3.6.2 Description of Architectural..................................................................29

3.7 Procedural/Modular Approach............................................................................30

3.7.1 Tour Page..................................................................………….........30

3.7.2 Tour Details Page.................................................................………..31

3.7.3 User Profile........................................................………....................32

3.8 Data Management..................................................................................32

3.9 Functional & Non- Functional /Operational Requirements......................33

Chapter – 4 Results & Discussions...........................................................................................35

4.1 Discussions..........................................................................................................35

4.2 Results..................................................................................................................36

4.2.1 Output Images.......................................................................................37

Chapter – 5 Conclusion.............................................................................................................39

References ................................................................................................................................40

Appendix...................................................................................................................................42

Structure of the Project File ………………………………………………………………….43

List Of Publications…………………………………………………………………………..45

# List of Tables

Table 1- Software Requirement……………………………………....16

Table 2- Hardware Requirements………………………………….....17

**List of Figures**

Fig.1:- Waterfall Model……………………………………………………....16

Fig.3.1:-MVC Architeccture……………………………………………….....25

Fig.3.2 :- Data Model Diagram………………………………….....................26

Fig.3.3: - JWT Authentication …………………………………...………...…27

Fig.3.4:- Strip Workflow…………………………………………………...…28

Fig.3.5:- Architecture Design………………………………………………....29

Fig.3.6:- Tours Design………………………………………………………...30

Fig.3.7:- Tour Details Page ……………………………………………….......31

Fig.3.8:- Use Profile page……………………………………………………..32

Fig.3.9: - Tour Page …………………………………………………………..36

Fig. 4.1: - Login Page………………………………………………………....37

Fig 4.2: - Account page……………………………………………………....38

Fig 4.3: - Payment Page……………………………………………………...38

# List of Symbols

• Circle Bullet

* Square Bullet

◦ Empty Bullet

* Three-D Top-Lighted Rightwards Arrowhead

# List of Abbreviations

**Terminology**  **Meaning**

ERD Entity Relationship Diagram

DBD Database Diagram

DFD Data Flow Diagram

UCD Use Case Diagram

HTTP Hyper Text Transfer Protocol

SQL Structure Query Language

DBMS Database Management System

IDE Integrated Development Environment

SDLC Software Development Life Cycle

CMS Content Management System

FAQ Frequently Asked Questions

***Chapter-I***

***Introduction***

Tourism In the ever-evolving landscape of web development, technologies such as Node.js, Express.js, and MongoDB have emerged as powerful tools, empowering developers to build robust and scalable applications. Within the domain of tour websites, these technologies offer a plethora of opportunities to create immersive digital experiences, revolutionizing the way travelers discover, plan, and embark on their adventures.

This thesis aims to delve into the exciting world of tour websites developed using Node.js, Express.js, and MongoDB, uncovering the immense potential that these technologies hold for web developers. By exploring the core features and functionalities of these frameworks and database, we can unveil the underlying architecture that drives the seamless user experience, real-time data processing, and efficient data management within tour websites.

Within the study, we will analyze the use of Node.js, a JavaScript runtime environment, and Express.js, a web application framework, in building the server-side components of tour websites. We will examine how the event-driven, non-blocking nature of Node.js allows for high-performance, scalable web applications, while Express.js simplifies the development process through its minimalistic and flexible approach. By understanding the intricacies of these technologies, developers can harness their power to create dynamic, responsive, and feature-rich tour websites.

Furthermore, this thesis will explore the integration of MongoDB, a NoSQL database, in tour websites, enabling developers to efficiently store and retrieve data. We will investigate how MongoDB's document-oriented structure, flexibility, and scalability make it an ideal choice for handling diverse data types, such as user profiles, travel itineraries, and destination information. By leveraging MongoDB's powerful querying and indexing capabilities, developers can optimize data access and provide users with a seamless browsing and search experience.

Through comprehensive research and analysis, this study aims to inspire web developers to embrace the potential of Node.js, Express.js, and MongoDB in the context of tour websites. We will explore the possibilities for leveraging these technologies to create personalized user experiences, implement real-time updates, and integrate advanced functionalities such as location-based services, social media integration, and recommendation systems.

Ultimately, this research seeks to unveil the untapped potential of tour websites built with Node.js, Express.js, and MongoDB, highlighting the opportunities for web developers to create innovative and transformative digital experiences. By recognizing the unique advantages of these technologies and their seamless integration, developers can unlock new avenues for enhancing user engagement, optimizing performance, and shaping the future of travel technology.

In conclusion, let us embark on this journey of web development, where the power of Node.js, Express.js, and MongoDB converges to empower the creation of captivating tour websites. By embracing these technologies and their inherent strengths, developers can push the boundaries of what is possible, revolutionizing the way travelers explore, plan, and embark on their journeys. Together, let us unveil the potential of this powerful tech stack and shape the future of tour websites.

***Chapter-II***

***Literature Review or Background Information***

**2.1 Background Information:**

* The Natours website, is a stunning example of a tour and travel website built using modern web development technologies and design principles. The website serves as a comprehensive platform for users to explore and book unique nature tours and adventures around the world.
* Design and User Experience: The Natours website boasts an exceptional design that perfectly captures the spirit of adventure and the beauty of nature. It features a visually appealing and immersive interface, characterized by vibrant colors, stunning imagery, and smooth animations. The website's user experience is carefully crafted, providing intuitive navigation, clear call-to-action elements, and engaging interactive features.
* Responsive and Mobile-Friendly: Recognizing the importance of mobile accessibility, the Natours website is fully responsive, adapting effortlessly to various screen sizes and devices. The responsive design ensures that users can access the website and browse tour options on smartphones, tablets, and desktops, allowing for a consistent and optimized experience across different platforms.
* Technologies and Architecture: Behind the scenes, the Natours website incorporates a range of modern web development technologies. The front-end development utilizes HTML5, CSS3, and JavaScript, employing advanced techniques such as flexbox and grid layouts for responsive design. The website also demonstrates the use of Sass, a CSS preprocessor, to streamline and organize the stylesheets.
* On the server-side, the Natours website relies on Node.js and Express.js to handle requests and serve dynamic content. These technologies allow for efficient routing, middleware management, and the implementation of server-side functionalities. MongoDB is utilized as the database of choice, enabling the storage and retrieval of tour data, user information, and booking details. The integration of Node.js, Express.js, and MongoDB ensures the website's scalability, performance, and robustness.
* Features and Functionality: The Natours website offers a range of features that enhance the user experience and facilitate tour exploration and booking. Users can browse through various nature tours, view detailed itineraries, and access stunning image galleries for each tour. The website provides a user-friendly booking process, allowing users to select preferred dates, customize tour options, and make secure online payments. Additionally, the website incorporates interactive maps to showcase tour locations, and it offers a review section where users can share their experiences and feedback.

# *Chapter-3*

# *Methodology or Method and Material*

**3.1 Introduction**

This Section describes the methodology applied during the development of NATOURS application. A methodology is a model, which project managers employ for the design, planning, implementation and achievement of their project objectives. Effective project management is essential in absolutely any organization, regardless of the nature of the business and the scale of the organization. From choosing a project to right through to the end, it is important that the project is carefully and closely managed.

The Waterfall Model was the first process model to be introduced by. It is also the earliest SDLC approach that was used for software development. That’s why this model is used to develop this project because of its simplicity. The waterfall model is divided into six phases which are Requirement Analysis, System Design, Implementation, Testing, Deployment, and maintenance. Waterfall approach also referred as Linear-Sequential Life Cycle Model where the outcome of a phase acts as the input for the next phase sequentially. That it is sequential design process as it seen as flowing steadily downwards like waterfall. This waterfall model must be completed by stage before moving to the next phase and there is no overlapping in the phases which means that any phases will only start if the previous one is completed.

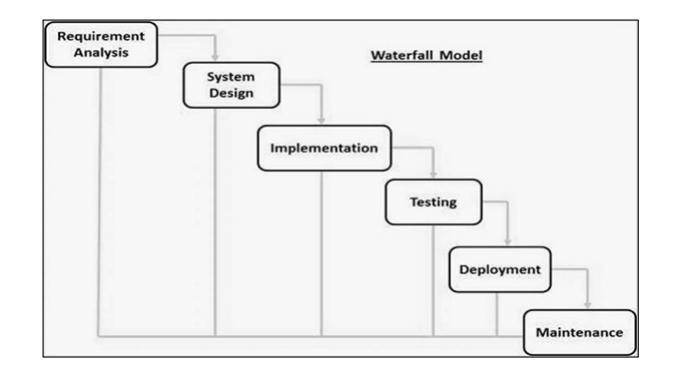


Fig.1: - Waterfall Model

**3.2 Justification for the Methodology**

When it comes to developing an Inventory Management System, selecting the right materials and methodology is crucial. The materials used to build an Inventory Management System must be reliable, scalable, and secure to ensure that the site functions properly and provides a positive user experience. The following are some of the materials and methods commonly used in Inventory Management App development:

1. Software as a service (SaaS) : - SaaS is a software distribution model in which a cloud provider hosts applications and makes them available to end users over the internet. In this model, an independent software vendor (ISV) may contract a third-party cloud provider to host the application. “Cloudinary” is a SaaS technology that provides cloud-based image and video management services. It enables users to upload, store, manage, manipulate, and deliver images and video for websites and apps. Cloudinary is used in this web-app to store the images being uploaded by the user.
2. Web hosting: - Web hosting is the service that allows a website to be published on the internet. When selecting a web host, it is important to choose a reliable provider that offers scalable solutions and provides high levels of security. Some popular web hosts include GitHub Pages, Vercel, and Render which are used to host this web-app.
3. PERN Stack :- PERN is one of several variations of the [MEAN stack](https://www.mongodb.com/mean-stack) (MongoDB Express Angular Node), where the traditional Angular.js front-end framework is replaced with Pug.js. Other variants include MEVN (MongoDB, Express, Vue, Node), and really any front-end JavaScript framework can work.

PERN stands for MongoDB, Express, PugJs, Node, after the four key technologies that make up the stack.

* MongoDB — MongoDB is a source-available cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas. MongoDB is developed by MongoDB Inc. and licensed under the Server Side Public License (SSPL) which is deemed non-free by several distributions. It’s main features are Ad-hoc Queries, Indexing, Replication, Load Balancing, File Storage, Aggregation, Transactions etc.
* Express(.js) — Express.js, or simply Express, is a back-end web application framework for building RESTful APIs with Node.js, released as free and open-source software under the MIT License. It is designed for building web applications and APIs. It has been called the de facto standard server framework for Node.js.
* PugJs (formerly known as Jade) is a popular template engine for Node.js and JavaScript web development. It is often used with Express.js, a web application framework for Node.js. Pug is designed to make writing HTML templates more concise and efficient. With Pug, you can write templates using indentation instead of explicit opening and closing tags. It supports features like variables, conditionals, loops, mixins, and includes, which allow you to create reusable and dynamic HTML templates. Pug templates are compiled into HTML before being sent to the client's browser. This makes it easier to manage and organize the structure of your web pages and promotes code reusability.
* Node(.js) — Node.js is an open-source server environment. Node.js is cross-platform and runs on Windows, Linux, Unix, and macOS. Node.js is a back-end JavaScript runtime environment. Node.js runs on the V8 JavaScript Engine and executes JavaScript code outside a web browser. Node.js lets developers use JavaScript to write command line tools and for server-side scripting. The functionality of running scripts server-side produces dynamic web page content before the page is sent to the user's web browser. Consequently, Node.js represents a "JavaScript everywhere" paradigm, unifying web-application development around a single programming language, rather than different languages for server-side and client-side scripts.

Like any web stack, you can build whatever you want in PERN — though it’s ideally suited for cases that are JSON-heavy, cloud-native, and that have dynamic web interfaces.

Examples include workflow management, news aggregation, to-do apps and calendars, and interactive forums/social products

**3.3 System Requirements and Analysis**

**3.3.1 System Analysis**

System analysis is the process of gathering and interpreting facts, diagnosing problems and using the information to recommend improvements on the system. System analysis is a problem - solving activity that requires intensive communication between the system users and system developers.

Analysis is an important part of any project; is analysis is not done properly then whole project move in the wrong direction. It also provides a schedule for proper project work. Analysis task divided into 3 areas:

* Problem Recognition.
* Feasibility Study.
* Requirement Analysis.

**3.3.2 System Feasibility**

Feasibility study of the system is a very important stage during system design. Feasibility study is a test of a system proposal according to its workability impact on the organization, ability to meet user needs, and effective use of resources. Feasibility study decides whether the system is properly developed or not. There are five types of feasibility as mentioned below:

1. Technical Feasibility

2. Time Schedule feasibility

3. Implementation feasibility

4. Economic Feasibility

1. Technical Feasibility: -

Technical feasibility corresponds to determination of whether it is technically feasible to develop the software. Here those tools are considered, which will be required for developing the project. The tools, which are available, and tools, which will be required, are taken into account. Considering all above points and aspects it is observed that the cost incurred in developing this project from a technical perspective would not be too high. Thus, it is feasible for company as well as for me to develop this system.

1. Time Feasibility: -

Time feasibility corresponds to whether sufficient time is available to complete the project. Parameters considered:

▪ Schedule of the project.

▪ Time by which the project has to be completed.

▪ Reporting period considering all the above factors it was decided that the allotted time that is 3 months was sufficient to complete the project.

1. Implementation Feasibility: -

Implementation Feasibility is about basic infrastructure required to develop the system. Considering all below points, it is feasible to develop system. Factors considered:

▪ All the minimum infrastructure facility required like PC, books, technical manuals are provided.

▪ Proper guidance is provided.

▪ All necessary data and files are provided.

1. Operational Feasibility: -

It refers to the feasibility of the product to be operational. Some products may work very well at the design and implementation but many fail in the real time environment. It introduces the study of human resources required and their technical expertise.

This product is operationally feasible as it is designed specifically for E-Governance. This provides consistent and integrated data management. It also provides information at all levels of people.

1. Economic Feasibility: -

Economic Feasibility is about total cost incurred for the system. The software resource requirement of the proposed system is Express.js and MongoDB for functional and backend development and React.js and CSS for the frontend UI.

**3.4. Requirements Analysis and Specification**

A complete understanding of software requirement is essential to the success of a web development effort. No matter how well designed or well coded, a poorly analyzed and specific program will disappoint user and bring grief to the developers.

The requirement analysis task is process of discovery, refinement, modified and specification. The software scope, initially established by the system engineer and refined during project planning, is refined in detail. Models of the required data, information and control flow, and operational behavior are created. Alternative solutions are analyzed and various project element.

We collected a number of requirements for project from our primitive research, website visits, and interview to the concerned personnel and their experiences regarding the concepts of its development. We have even visited some organization in Chhattisgarh and analyze its importance and try to develop the project by fulfilling all the weakness that were found in the application. We then decided to build same type of application with different logic flow and new language which will be suitable for the small organization.

Usually, creating inventory list can be done two ways either manually or computerized. Manually inventorying is usually written data with handwriting on the data books. It has many disadvantages. One of them is when the admin staff input the data, it is prone to be typographical errors. This app will reduce the efforts of maintaining inventory stocks manually and it can be done more easily with multiple features and customer can maintain the more secured way and less prone to errors.

**3.4.1. Requirements Gathering: -**

Requirement Gathering also known as data collection. Data Collection is an important aspect of any type of research study. Inaccurate data collection can impact the results of a study and ultimately lead to invalid results. The methods used to gather the projects requirements involves Quantitative research to review the existing systems in the market.

**3.4.2. HARDWARE REQUIREMENT:**

|  |  |  |  |
| --- | --- | --- | --- |
| S No. | Hardware | Description | Functionality |
| 1. | Laptop | * Processor:   3.4 GHz Intel Core i3  RAM: 4GB  HDD/SSD: 128GB | * Use to develop the Inventory Management web-app. |
| 2. | Printer | * Model:   HP Deskjet 2135 Series | * To print the documentation of the project   proposal. |
| 3. | Hard Disk | * Model:   Seagate Portable Drive | * To back-up the project. |

**Table 1**: Hardware Requirement

**3.4.3. SOFTWARE REQUIREMENT:**

**Table 2**: Software Requirement

|  |  |  |
| --- | --- | --- |
| S No. | Software | Description |
| 1. | MongoDB | * System database application. |
| 2. | Sublime & VS Code | * Implementation of coding. |
| 4. | Microsoft Word for Windows | * Preparation of the documentation of project proposal. |
| 5. | Microsoft PowerPoint for  Windows | * Preparation of the presentation of the project proposal. |
| 6. | Microsoft Office | * Design of the ERD, UCD and DFD. |

**3.5. System Design**

The section describes the system study, analysis, design strengths and weaknesses of the current system, Contest level diagrams, Entity Relationship Diagram, Architectural design. After interpretation of the data, tables were drawn and process of data determined to guide the researcher of the implementation stage of the project. The tools, which were employed during this methodology stage, where mainly tables, Data Flow Diagrams and Entity Relationship Diagrams. The design ensures that only allows authorized users to access the systems information.

**3.5.1 Design Goals**

The following goals were kept in mind while designing the system:

* Make system user-friendly. This was necessary so that system could be used efficiently and system could act as catalyst in achieving objectives.
* Make system compatible i.e. It should fit in the total integrated system. Future maintenance and enhancement must be less.
* Make the system compatible so that it could integrate other modules of system into itself.
* Make the system reliable, understandable and cost-effective

**3.5.2 MVC Architecture**

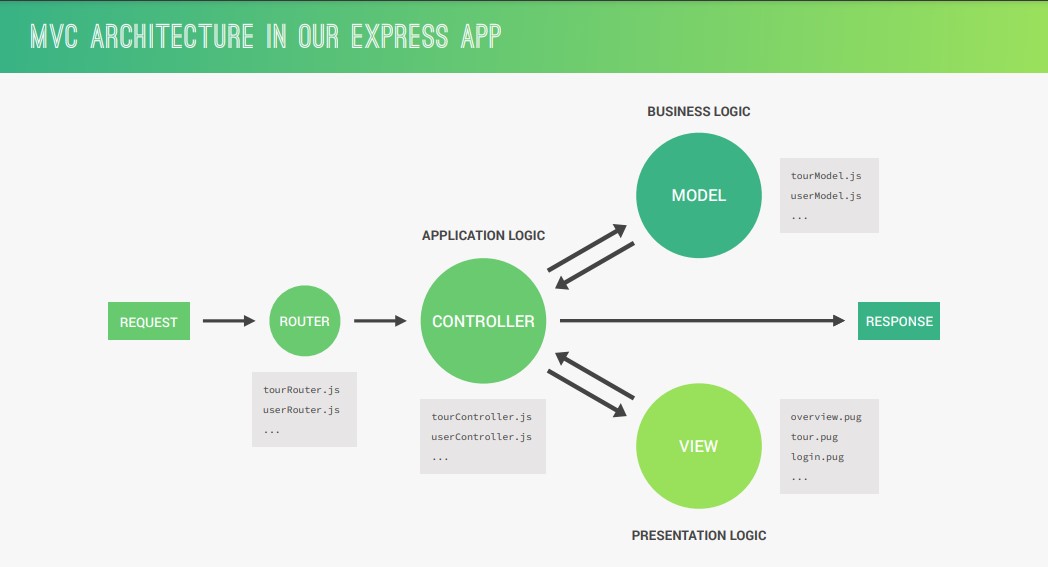


Fig.3.1:- MVC Architecture

**3.5.3 Data Model Diagram**

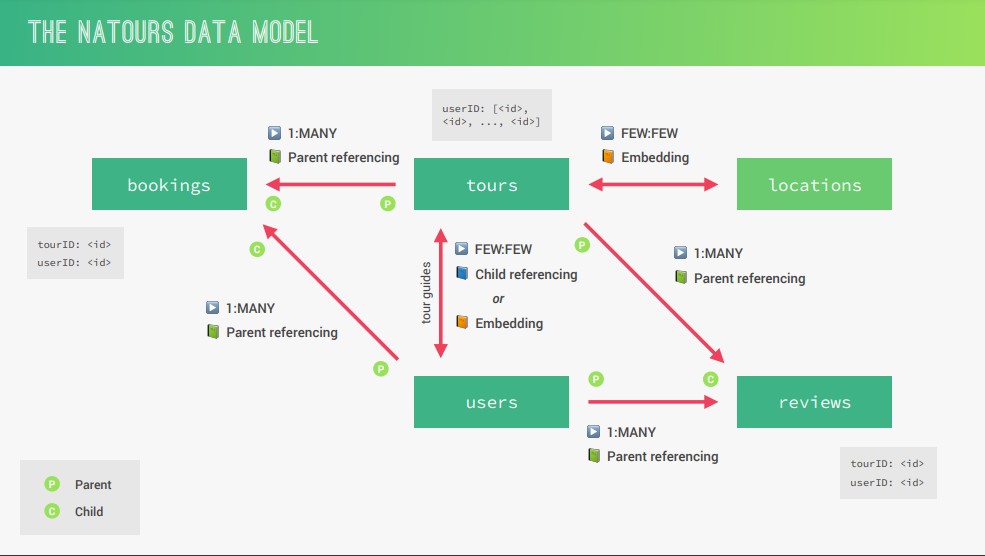


Fig.3.2:- Data Model Diagram

**3.5.4 JWT Authentication**

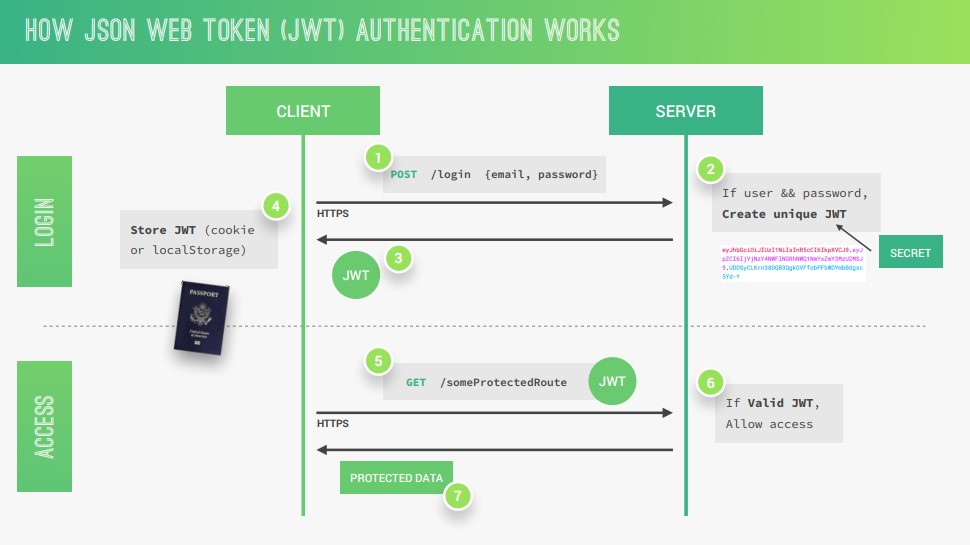


Fig.3.3:- JWT Authentication

**3.5.3 Stripe Workflow**

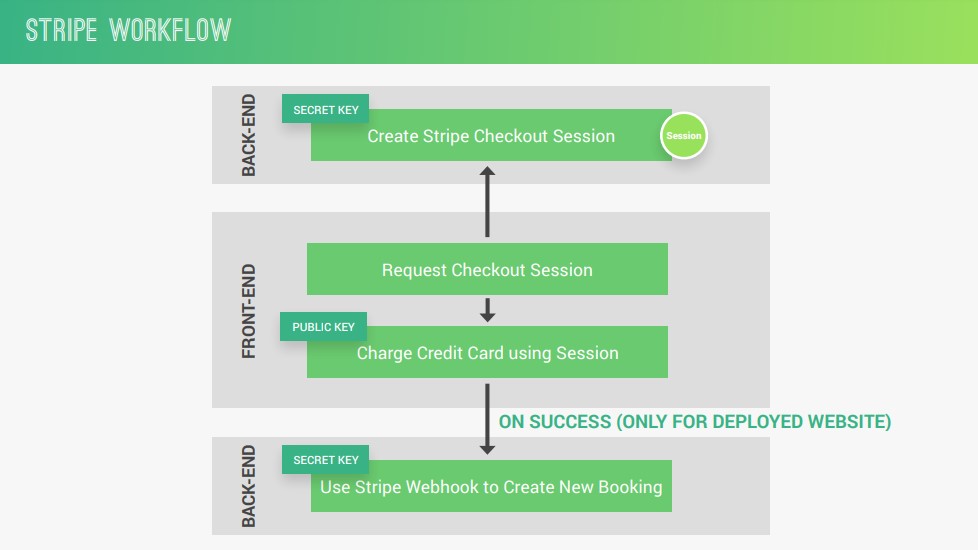


Fig.3.4:- Stripe Workflow

**3.6 Architectural Design**

3.6.1 Architectural Diagram

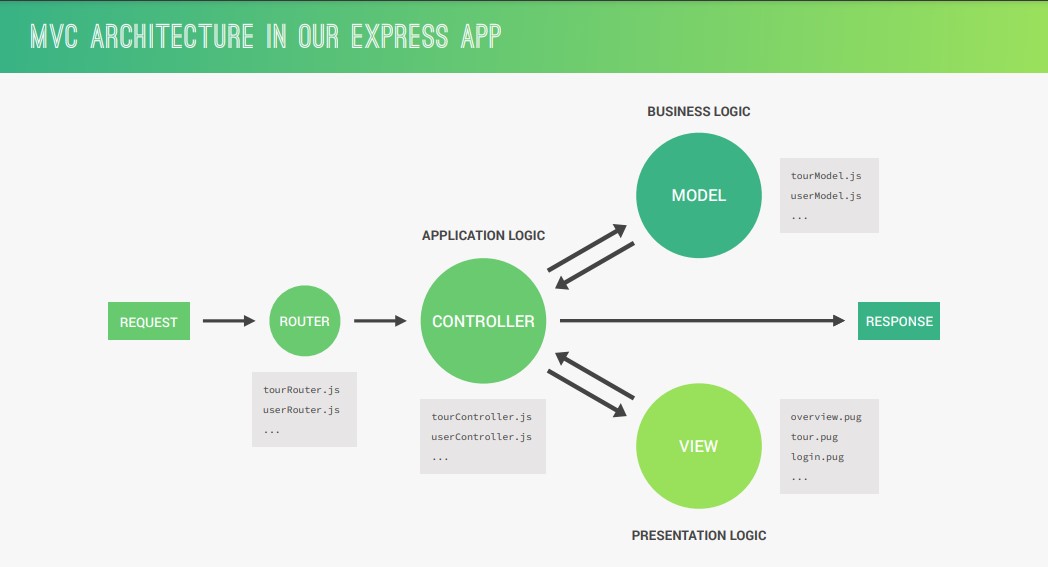


Fig.3.6:- Architectural Diagram

**3.6.2 Description of Architectural Design**

Natours website utilizes HTML, CSS, and JavaScript to create the user interface and interactive elements. Advanced CSS techniques, such as flexbox and grid layouts, are employed to achieve responsive and visually appealing designs. Additionally, the use of CSS preprocessors like Sass allows for modular and organized stylesheets. JavaScript is used to enhance the interactivity of the website, handle user interactions, and facilitate dynamic content updates. The back-end of the Natours website is built using a Node.js and Express.js framework. Node.js is a server-side JavaScript runtime that enables the execution of JavaScript on the server. Express.js, a web application framework for Node.js, provides a robust and flexible foundation for building server-side logic, handling routing, and managing middleware.

**3.7 Procedural/Modular Approach**

Following are all the modules designed for the Natours App.

**3.7.1 Tours Page**

The home page is the main entry point of the website. It typically features stunning imagery, captivating headlines, and a brief introduction to the website's offerings. The home page may also highlight popular tours, special promotions, or featured destinations to engage and entice visitors.

.

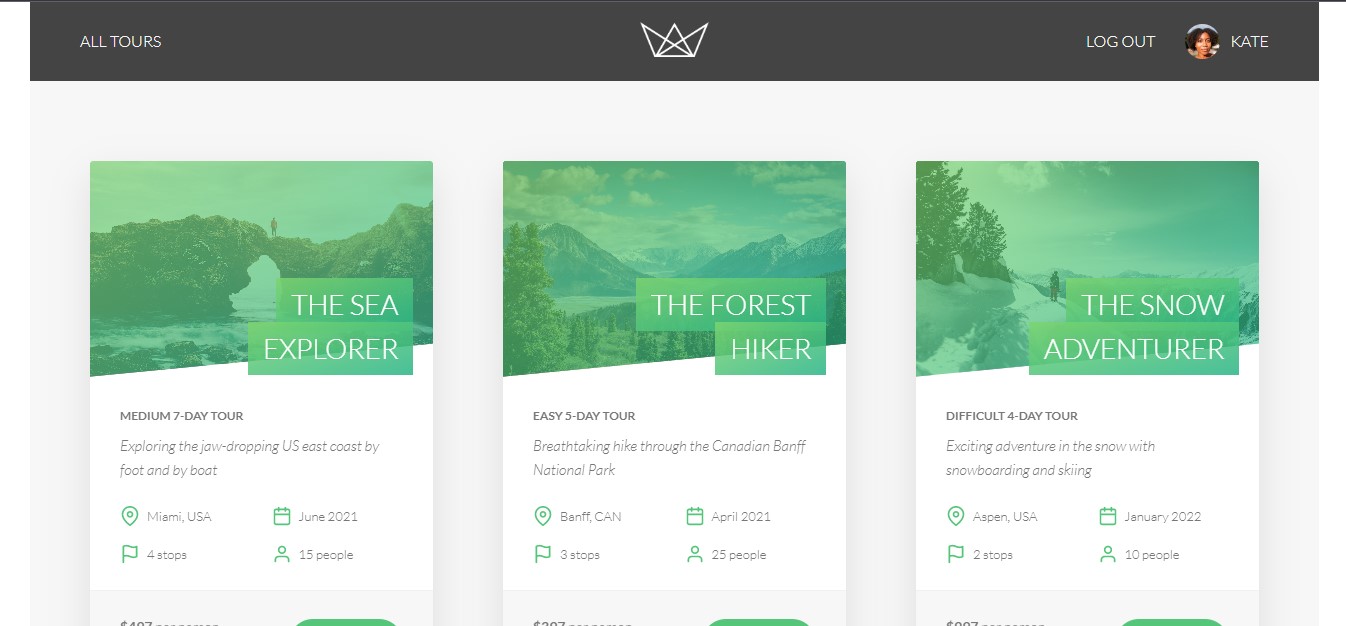


Fig.3.7: - Tours Page

**3.7.2 Tour Details Page**

The tour details page provides comprehensive information about a specific tour. It includes a detailed itinerary, showcasing the day-by-day activities and locations covered during the tour. The page may also display high-quality images or videos related to the tour, allowing users to visualize the experience. Additionally, users can find essential details such as the tour's difficulty level, included amenities, departure dates, and any special requirements.

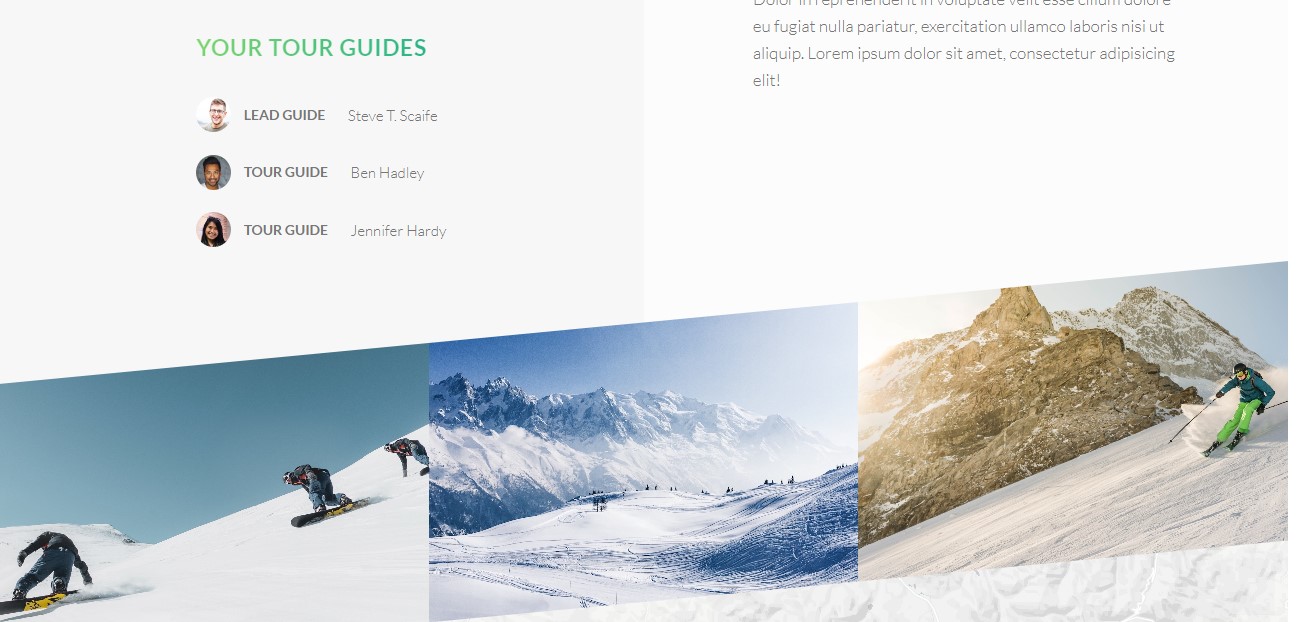


Fig.3.8:- Tour Details Page

**3.7.3 User Profile Page**

The user profile page provides registered users with a personalized dashboard. Here, users can view and manage their bookings, review their past and upcoming tours, update their contact information, and possibly access any loyalty rewards or membership benefits. The profile page may also provide recommendations based on the user's previous bookings or preferences.

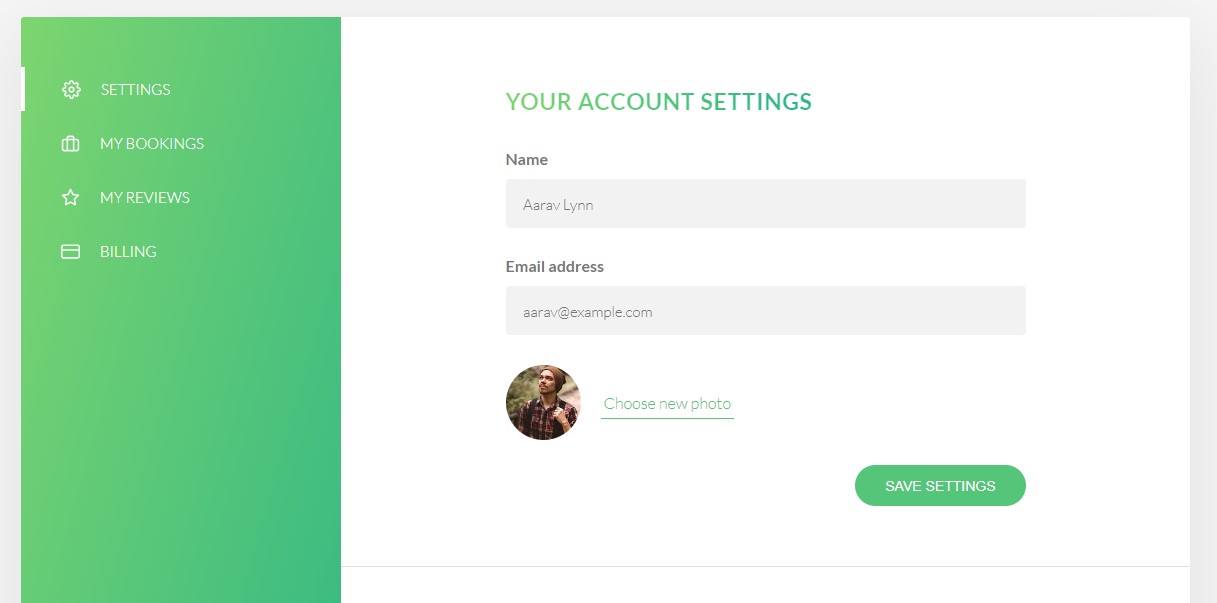


Fig.3.9:- User Profile Page

**3.8 Data Management**

* **Data Description**

This database consists of

* Users: User’s information is added to database with Unique ID.
* Tours: Complete tours information is stored in this collection.
* Tokens: Customer when gets logged in to their account are assigned a token which lasts for 24 hours so that their session automatically expires after 24 hours.
* **Data Objects**
* User: ID, User Name, Password, Email, Photo, Phone Number, Bio, Created At, Updated At
* Products: ID, User, Product Name, SKU, Price, Quantity, Category, Image, Description, Created At, Updated At.
* Tokens: ID, User Id, Token, Created At, Expires At.

**3.9 FUNCTIONAL & NON-FUNCTIONAL / OPERATIONAL REQUIREMENTS:**-

**Functional Requirements**

The following is the desired functionality of the new system. The proposed project would cover:

**User Module**

• After creating account and login to system, user can create their own account.

• User can view tours.

• User can also add/remove there account data.

• User can upload the image for there profile.

**Non-functional Requirements:-**

It specifies the quality attribute of a software system. They judge the software system based on Responsiveness, Usability, Security, Portability and other non-functional standards that are critical to the success of the software system.

**• Availability:** The system should remain operational in any day and any place.

• **Accuracy:** There is a need to optimize the system to ensure more accurate results and calculations.

• **Usability:** The system should provide a User-friendly user interface and tooltips to enhance itself and be effectively responsive.

• **Secure:** The system must be able to provide security against any external injections by using a layered security system. Implementation of user login functionalities also ensures the system is secure from unauthorized persons.

• **Performance of the system:** Response time is very good for given piece of work. The system will support multi user environment.

• **Reliability of the system:** The system will be highly reliable and it generates all the updates information in correct order. Data validation and verification is done at every stage of activity.

***Chapter-IV***

***Results & Discussion***

**4.1 Discussion:**

The output of the Natours web application, is a visually stunning and user-friendly platform that delivers an immersive and engaging experience for users interested in nature tours and adventures. The web application combines captivating design elements, interactive features, and a seamless user interface to provide a range of outputs that cater to different user needs.

One of the key outputs of the Natours web application is the ability for users to explore and browse a diverse selection of nature tours. The website presents users with detailed information about each tour, including descriptions, itineraries, durations, and prices. Through visually appealing imagery and engaging content, users can gain a comprehensive understanding of the tours available, making it easier for them to select the experiences that match their preferences and interests.

Another significant output of the web application is the seamless booking process. Users can customize their tour options, select preferred dates, specify the number of participants, and even choose optional add-ons or accommodations. The web application provides a user-friendly booking form that guides users through the process, collecting necessary information and facilitating secure online payments. This output allows users to effortlessly reserve their desired tours and secure their bookings with confidence.

The Natours web application also generates outputs that enhance user engagement and facilitate informed decision-making. The inclusion of user-generated reviews and ratings for tours enables prospective users to access feedback from fellow travelers, assisting them in making well-informed choices. By providing a platform for users to share their experiences and opinions, the web application fosters a sense of community and trust among its users.

Furthermore, the web application outputs an intuitive user profile section where registered users can manage their bookings, review past and upcoming tours, update their contact information, and access any loyalty rewards or membership benefits. This personalized output ensures that users have a convenient and centralized hub to track their tour-related activities and preferences.

**4.2 Results:**

After using this software, we have got some result that is-

* Outstanding performance for business calculation and logic
* Robust, secure & portable
* Easy to use & User friendly
* Customizable
* Integrity feature
* Smart reporting system of invoice
* Dynamic and nice look & feel of UI
* Fast Performance
* Data backup and restore system

**4.2.1 Output Images:**

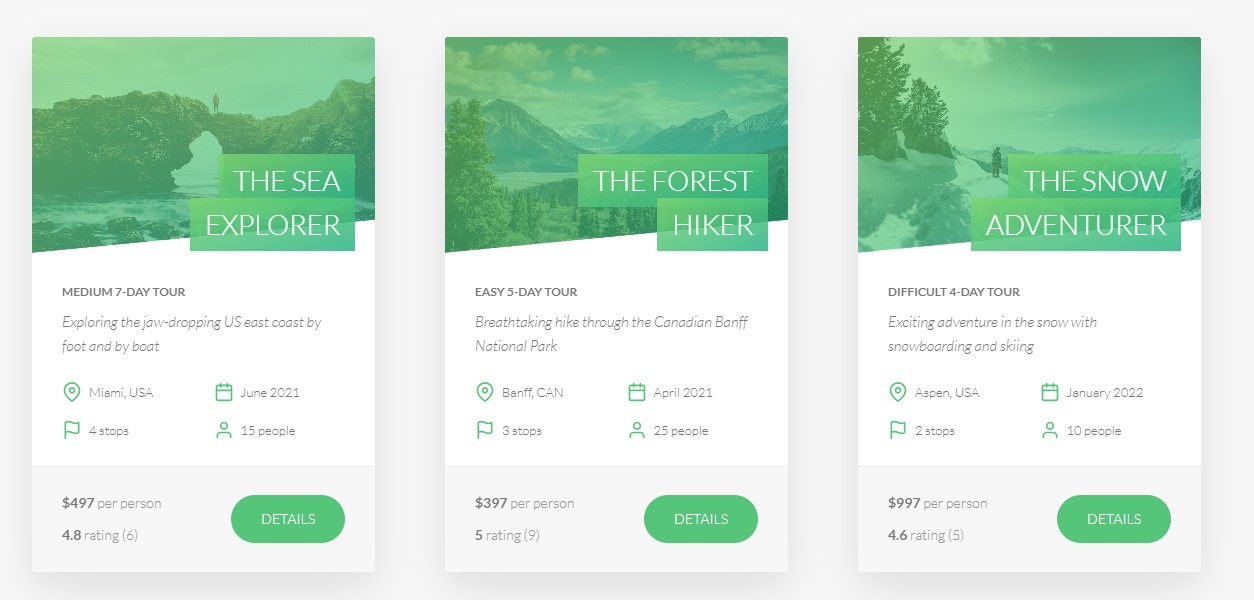


Fig 4.1 – Tours page

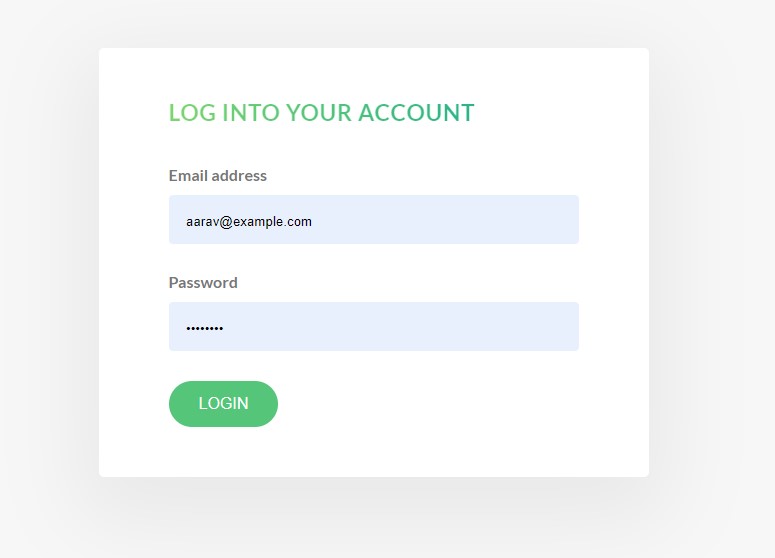


Fig 4.2- Login Page

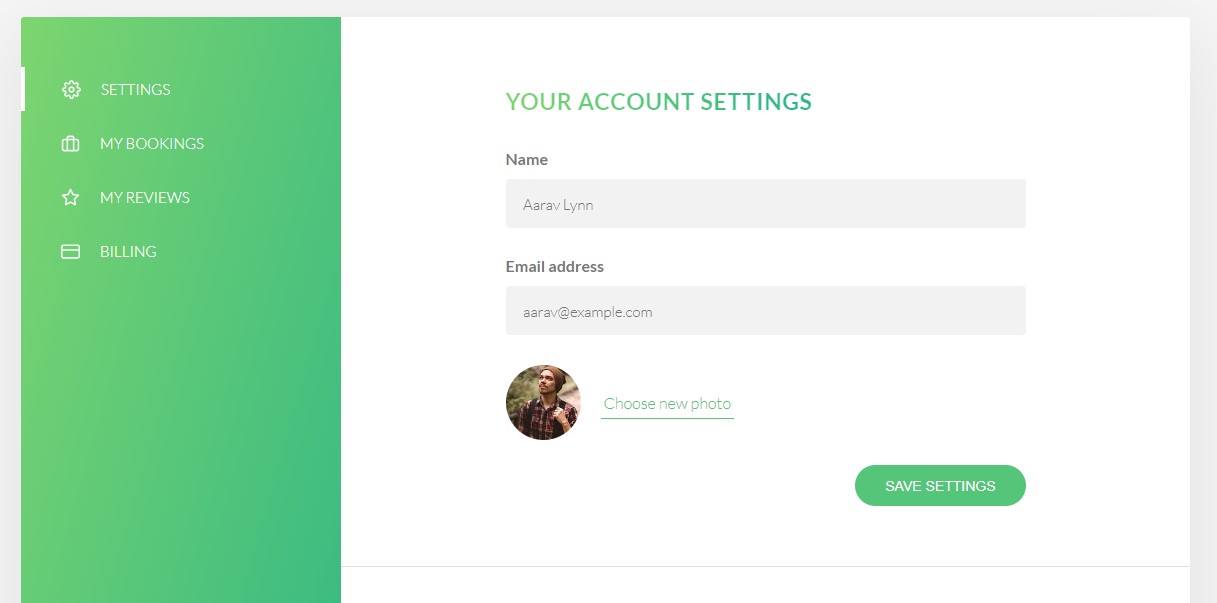


Fig 4.3 – Account Page

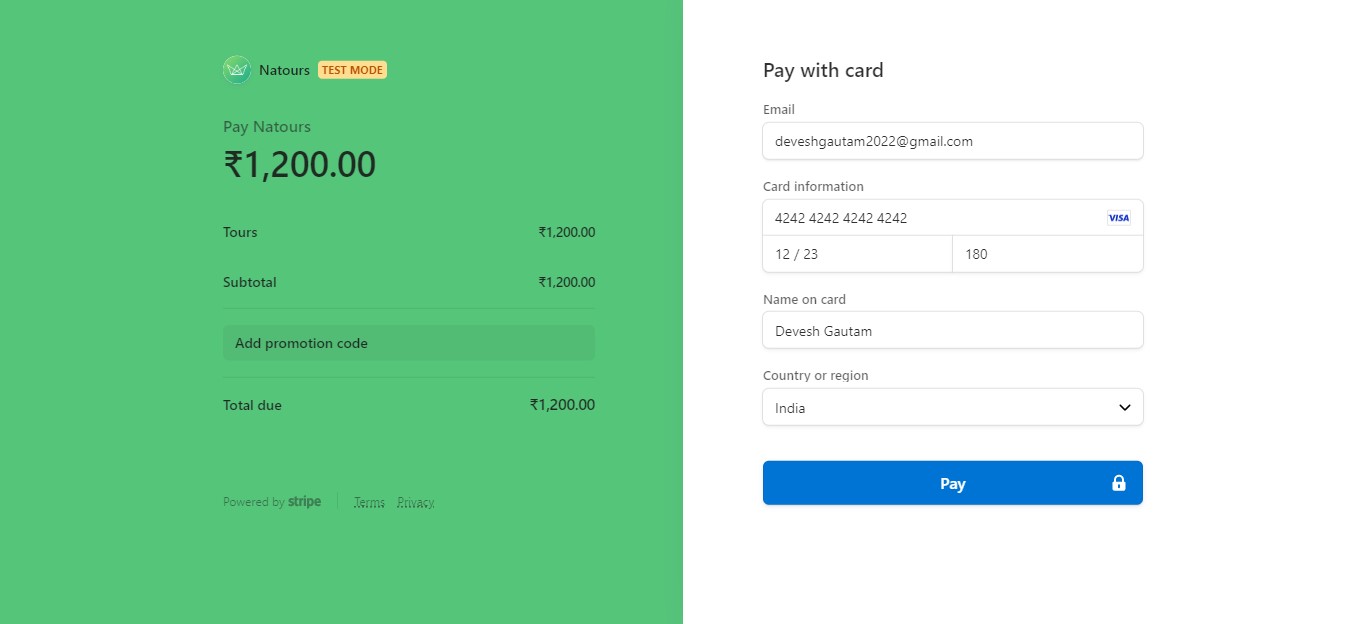


Fig 4.4 – Payment page

***Chapter-V***

***Conclusion***

**Conclusion:**

The main objective of the project is to develop a Secure Generic Web-App which can morph easily into any kind of management system. We have taken a wide range of literature review in order to achieve all the tasks, where we came to know about some of the products that are existing in the market. We made detailed research in that path to cover the loop holes that existing systems are facing and to eradicate them in our application. In the process of research, We came to know about the latest technologies and different algorithms.

**Future Scope:**

* Cross Platform App
* Cloud Enabled
* Chat Bot
* Used as a Template
* Query resolution
* Compatibility IOT Devices

# References

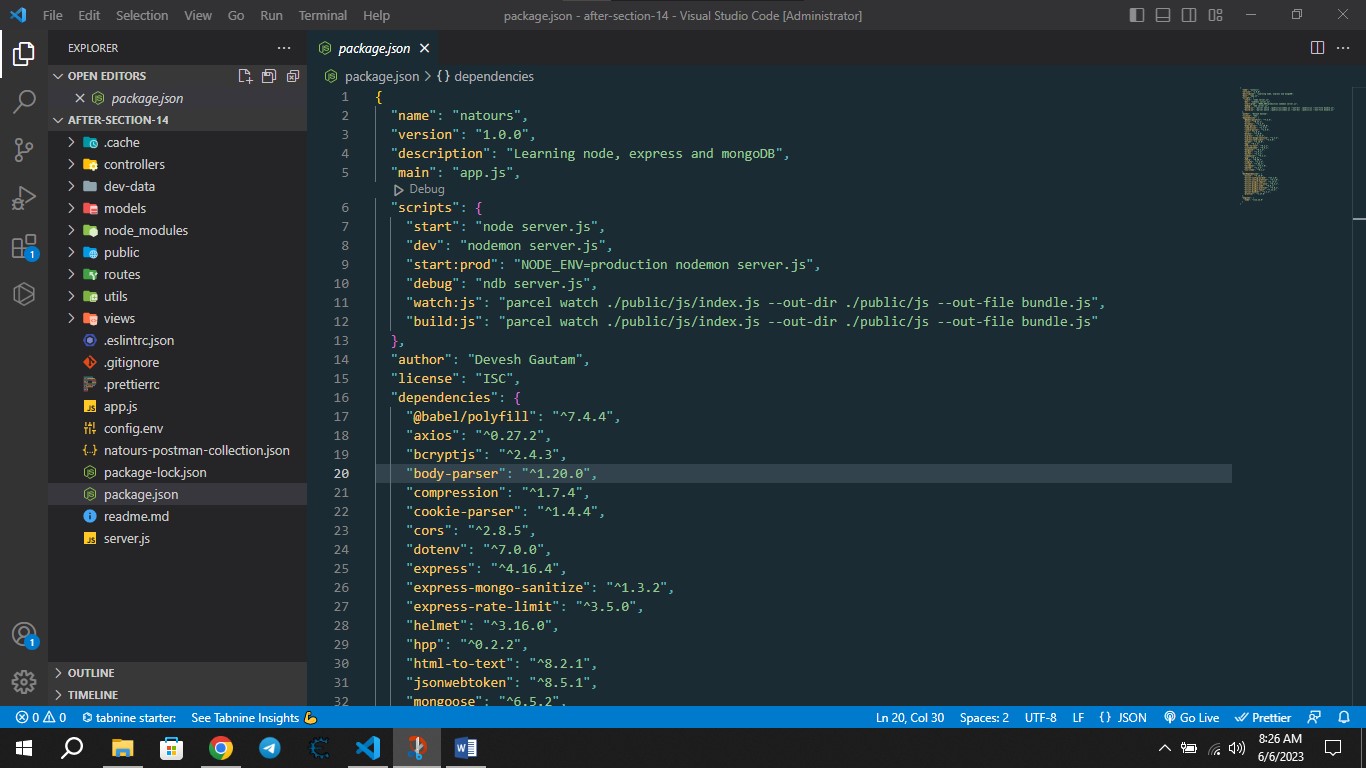
* Aaker, D. A. (2010). Building strong brands. Simon and Schuster.
* Alrick, P. L., & Settle, R. B. (2004). The survey research handbook (3rd ed.). McGraw-Hills.
* Baker, M. J., & Hart, S. (2007). The marketing book (6th ed.). Butterworth-Heinemann.
* Kotler, P., & Armstrong, G. (2010). Principles of marketing (13th ed.). Prentice Hall.
* "React: Up & Running: Building Web Applications" by Stoyan Stefanov and Adam Freeman.
* Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React, and Node" by Vasan Subramanian.
* <https://www.mongodb.com/home>
* <https://expressjs.com/>
* <https://reactjs.org/>
* <https://nodejs.org/en/>
* https://[www.wikipedia.com](http://www.wikipedia.com)/
* <https://www.youtube.com/>

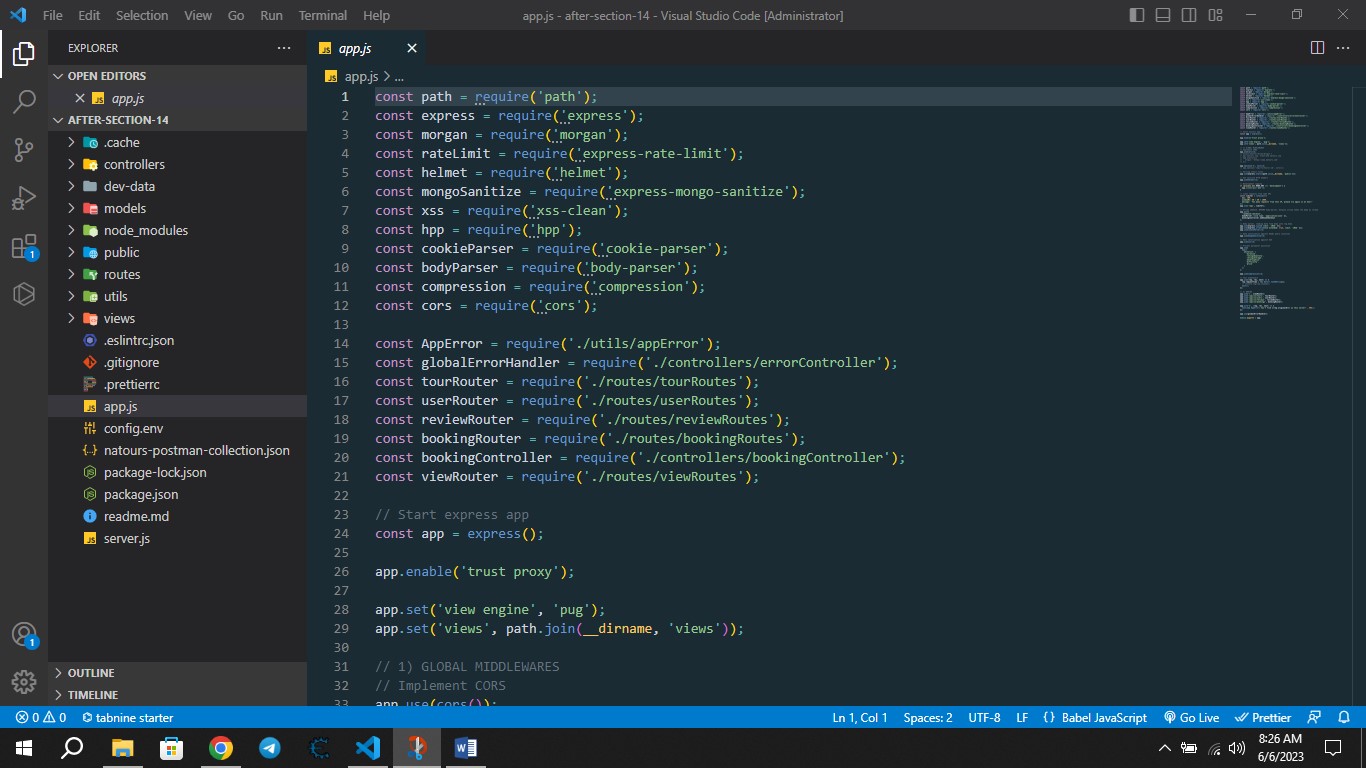
# Appendix

Record of all procedures and Results

|  |  |
| --- | --- |
| **Registration** | This procedure allows the customers to sign in or sign up in order to create their inventory and store items in it. |
| **Product** | This module contains the details of all products in the store. |
| **Price** | Manages Prices of products. |
| **Actions** | Manages the actions performed by the user on the items in the inventory. |
| **Payment** | Customer makes payment at this stage. However, this system does not have a real payment system. |
| **Search** | Customers are able to search for specific product through this procedure |
| **Layout** | Customers view products from the dashboard. They can Mouse over to display product description. |
| **Reporting** | The system also provides room form customer feedback. |

# Structure of the Project File:

****

****

Roach website is a Model View Control (MVC) framework:-

* public/: This directory contains static assets that are directly accessible by the client-side, such as CSS stylesheets (css/), images (img/), and JavaScript files (js/).
* src/: This directory holds the main source code of the application.
* controllers/: This directory contains the logic and functionality of the application, typically organized as separate modules or controllers.
* models/: This directory houses the database models or schemas used in the application.
* routes/: This directory contains the route definitions and handlers for handling different HTTP requests.
* views/: This directory contains the templates and views used to generate dynamic HTML pages
* includes/: This directory contains reusable components or partials that can be included in multiple views.
* layouts/: This directory holds the base layout templates that define the overall structure of the pages.
* pages/: This directory contains the individual page templates.
* app.js: This file is the entry point of the application, where the server setup, middleware configuration, and routing are typically defined.
* .env: This file stores environment variables used in the application, such as database connection details or API keys.
* .gitignore: This file specifies which files or directories should be ignored by version control systems like Git.
* package.json and package-lock.json: These files contain metadata about the project and its dependencies.
* README.md: This file typically includes documentation, instructions, and information about the project.

**List of Publication**

1. R. Gupta, S. Sharma, "E-Commerce Websites Design and Development: A Review," International Journal of Computer Applications, vol. 136, no. 5, pp. 1-6, Feb. 2016.
2. S. Saini, S. Bansal, "An Analysis of E-Commerce Websites Using Web Mining Techniques," International Journal of Computer Science and Information Technology, vol. 6, no. 4, pp. 148-157, Aug. 2014.
3. H. Choi, J. Lee, "Exploring the Effect of Online Customer Reviews on E-Commerce Websites: An Empirical Study," Journal of Internet Commerce, vol. 14, no. 3, pp. 1-19, Aug. 2015.
4. S. Choudhary, R. Kaur, "Factors Affecting Consumer Trust on E-commerce Websites: A Literature Review," Journal of Emerging Technologies and Innovative Research, vol. 6, no. 7, pp. 399-406, Jul. 2019.
5. K. Jain, A. Sharma, "A Study on Customer Buying Behaviour on E-commerce Websites in India," International Journal of Management and Social Sciences Research, vol. 6, no. 5, pp. 32-39, May 2017.
6. S. Mehta, A. Singh, "E-commerce Websites: An Exploratory Study of Factors Affecting Consumer Behaviour in India," Journal of Business and Retail Management Research, vol. 12, no. 3, pp. 87-99, Apr. 2018.
7. P. Singh, N. Srivastava, "A Study on the Effectiveness of Social Media Marketing on E-commerce Websites," International Journal of Research in Management, Science and Technology, vol. 7, no. 2, pp. 8-15, Feb. 2017.
8. S. Singh, S. Chauhan, "Challenges and Opportunities in E-Commerce Website Design and Development," International Journal of Emerging Technology and Advanced Engineering, vol. 5, no. 7, pp. 329-335, Jul. 2015.
9. P. Sharma, P. Aggarwal, "E-commerce in India: Current Status and Future Prospects," International Journal of Business and Management Studies, vol. 7, no. 1, pp. 35-42, Jan. 2015.
10. S. Garg, S. Aggarwal, "A Review of E-Commerce Website Development Platforms," International Journal of Advanced Research in Computer Science and Software Engineering, vol. 8, no. 5, pp. 252-258, May 2018.