**ABSTARCT**

As Tourism is one of the fastest growing industries today, thus within the tourism industry events are getting more and more important. People have become more interested in events of all kinds, and will travel far away to participate in events that they find interesting. Events can offer various economical and social benefits for destinations, and therefore destination managers can and should employ events effectively in tourism role. It has become widely accepted that every community and destination needs to adopt a long term, strategic approach to event tourism there by planning and development in order to realize the full tourism potential of events.

Tourism application is a unique and specialized website (portal) on tour and travelling. The website is highly useful for users or tourists. This app provides information about various places of south India. With the help of our website user can view the desired places, user has the option to select various packages . After selecting the desired package, he can entire the payment details such as card number, cvv number, expiration date and can book his desired destination thus save his time instead of waiting in queue at the stations.

**INTRODUCTION**

**HTML**

Hypertext Markup Language (HTML) is the standard markup language for creating web pages and web applications. With Cascading Style Sheets (CSS) and JavaScript it forms a triad of cornerstone technologies for the World Wide Web. Web browsers receive HTML documents from a web server or from local storage and render them into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects, such as interactive forms, may be embedded into the rendered page. It provides a means to create structured documents by denoting structural semantics for text  
such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets. Tags such as <img /> and <input /> introduce content into the page directly. Others such as <p>...</p> surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

HTML can embed programs written in a scripting language such as JavaScript which affect the behavior and content of web pages. Inclusion of CSS defines the look and layout of content.

**CSS**

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language. Although most often used to set the visual style of web pages and user interfaces written in HTML and XHTML, the language can be applied to any XML document, including plain XML, SVG and XUL, and is applicable to rendering in speech, or on other media. Along with HTML and JavaScript, CSS is a cornerstone technology used by most websites to create visually engaging webpages, user interfaces for web applications, and user interfaces for many mobile applications.

CSS is designed primarily to enable the separation  
of presentation and content, including aspects such as the layout, colors, and  
fonts. This separation can improve content accessibility, provide more  
flexibility and control in the specification of presentation characteristics, enable multiple HTML pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content.

**JAVASCRIPT**

JavaScript (sometimes abbreviated JS) is a prototype-based scripting language that is dynamic, weakly typed, general purpose programming language and has first-class functions. It is a multi-paradigm language, supporting object-oriented, imperative, and functional programming styles.

JavaScript was formalized in the ECMA Script language standard and is primarily used in the form of client-side JavaScript, implemented as part of a Web browser in order to provide enhanced user interfaces and dynamic websites. This enables programmatic access to computational objects within a host environment.

JavaScript's use in applications outside Web pages for example in PDF documents, site-specific browsers, and desktop widgets is also significant.

In this application, JavaScript is used for validation purpose like text box validation, email validation, phone number validation. JavaScript is the good tool for validating the web-applications.

**PYTHON FLASK**

Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries.It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling various open authentication technologies and several common framework related tools. Extensions are updated far more regularly than the core Flask program.

Components

The microframework Flask is based on the Pocoo projects Werkzeug and Jinja2 WerkzeugWerkzeug is a utility library for the Python programming language, in other words a toolkit for Web Server Gateway Interface (WSGI) applications, and is licensed under a BSD License. Werkzeug can realize software objects for request, response, and utility functions. It can be used to build a custom software framework on top of it and supports Python 2.6, 2.7 and 3.3.[14]

Jinja

Jinja is a template engine for the Python programming language and is licensed under a BSD License also by Ronacher. Similar to the Django web framework, furthermore it provides that templates are evaluated in a sandbox.

**SQLITE**

SQLite is a relational database management system contained in a C programming library. In contrast to many other database management systems, SQLite is not a client–server database engine. Rather, it is embedded into the end program.SQLite is ACID-compliant and implements most of the SQL standard, using a dynamically and weakly typed SQL syntax that does not guarantee the domain integrity.SQLite is a popular choice as embedded database software for local/client storage in application software such as web browsers. It is arguably the most widely deployed database engine, as it is used today by several widespread browsers, operating systems, and embedded systems (such as mobile phones), among others. SQLite has bindings to many programming languages.

SQLite implements most of the SQL-92 standard for SQL but it lacks some features. For example, it partially provides triggers, and it cannot write to views (however it provides INSTEAD OF triggers that provide this functionality). While it provides complex queries, it still has limited ALTER TABLE function, as it cannot modify or delete columns.

SQLite uses an unusual type system for an SQL-compatible DBMS; instead of assigning a type to a column as in most SQL database systems, types are assigned to individual values; in language terms it is dynamically typed. Moreover, it is weakly typed in some of the same ways that Perl is: one can insert a string into an integer column (although SQLite will try to convert the string to an integer first, if the columns preferred type is integer). This adds flexibility to columns, especially when bound to a dynamically typed scripting language. However, the technique is not portable to other SQL products. A common criticism is that SQLite's type system lacks the data integrity mechanism provided by statically typed columns in other products. The SQLite web site describes a "strict affinity" mode, but this feature has not yet been added. However, it can be implemented with constraints like CHECK(typeof(x)='integer'). Tables normally include a hidden rowid index column which gives faster access. If a database includes an Integer Primary Key column SQLite will typically optimize it by treating it as an alias for rowid, causing the contents to be stored as a strictly typed 64-bit signed integer and changing its behavior to be somewhat like an auto-incrementing column. Future versions of SQLite may include a command to introspect whether a column has behavior like that of rowid to differentiate these columns from weakly-typed, non-auto incrementing Integer Primary Keys.Several computer processes or threads may access the same database concurrently. Several read accesses can be satisfied in parallel. A write access can only be satisfied if no other accesses are currently being serviced. Otherwise, the write access fails with an error code (or can automatically be retried until a configurable timeout expires). This concurrent access situation would change when dealing with temporary tables. This restriction is relaxed in version 3.7 when write-ahead logging (WAL) is turned on enabling concurrent reads and writes. SQLite version 3.7.4 first saw the addition of the FTS4 (full text search) module, which features enhancements over the older FTS3 module. FTS4 allows users to perform full text searches on documents similar to how search engines search webpages. Version 3.8.2 added support for creating tables without rowid, which may provide space and performance improvements. Common table expressions support was added to SQLite in version 3.8.3.

Development and distribution

SQLite's code is hosted with Fossil, a distributed version control system that is itself built upon an SQLite database.A standalone command-line program is provided in SQLite's distribution. It can be used to create a database, define tables, insert and change rows, run queries and manage an SQLite database file. It also serves as an example for writing applications that use the SQLite library.SQLite uses automated regression testing prior to each release. Over 2 million tests are run as part of a release's verification. Starting with the August 10, 2009 release of SQLite 3.6.17, SQLite releases have 100% branch test coverage, one of the components of code coverage. The tests and test harnesses are partially public domain and partially proprietary.

Notable users

Middleware

ADO.NET adapter, initially developed by Robert Simpson, is maintained jointly with the SQLite developers since April 2010.

ODBC driver has been developed and is maintained separately by Christian Werner. Werner’s ODBC driver is the recommended connection method for accessing SQLite from OpenOffice.org.

COM (ActiveX) wrapper making SQLite accessible on Windows to scripted languages such as JScript and VBScript. This adds SQLite database capabilities to HTML Applications (HTA).XULRunner uses SQLite

Web browsers

The browsers Google Chrome, Opera, Safari and the Android Browser all allow for storing information in, and retrieving it from, a SQLite database within the browser, using the Web SQL Database technology, although this is rapidly becoming deprecated (namely superseded by IndexedDB).

Mozilla Firefox and Mozilla Thunderbird store a variety of configuration data (bookmarks, cookies, contacts etc.) in internally managed SQLite databases.

**CHAPTER 2**

**REQUIREMENT ANALYSIS**

**HARDWARE REQUIREMENTS**

The Hardware requirements are very minimal and the program can be run on most of the machines.

|  |  |  |
| --- | --- | --- |
| Processor | : | Intel core i3. |
| Processor Speed | : | 2.4 GHz |
| RAM | : | 4 GB |
| Storage Space | : | 40 GB |
| Monitor Resolution | : | 1024\*768 or 1336\*768 or 1280\*1024 |

**SOFTWARE REQUIREMENTS**

1. Operating System used: Ubuntu 18.04
2. Frontend tool: HTML, CSS, JavaScript.
3. Interface: Python flask.
4. Backend tool: SQLite database.

**FUNCTIONAL REQUIREMENTS**

**2.3.1 Major Entities**

* Home
* Packages
* Booking
* Payment
* Back

**End User Requirement**

The user can login to website and can view the details of places and packages and can book a travel to desired place. The website can also be opened in cellphones.

**DATABASE**

**(SQLITE)**

**WEB BROWSER**

**CLOUD**

**(AWS)**

**FFFINTERFACE(FLASK)**

**USER**