



EVENT PLANNING

WEDDINGS/PARTIES/BUSINESS

UNIVERSITY OF KARACHI
DEPARTMENT OF COMPUTER SCIENCE

“Event Planner”

Name: MAH-E-FALAK	Name: AMNA LATIF
Seat No# EP - 1550022	Seat No# EP - 1550008
Class: BSSE – 3 rd Year (Evening)	Class: BSSE – 3 rd Year (Evening)
Section: A	Section: A

Internet Programming Techniques
(502)

Submission Date: 21st December, 2017

Submitted To: Sir Zohaib Najam

INDEX

S.No	Contents
1.	Project Proposal
2.	Mockup
3.	Project Description
4.	API Document
5.	README.MD file
6.	Project Complete Code Link
7.	ERD Diagram

Event Planner Website

Description:

Our Project is to design an Event Planner Website. The purpose of this website is to serve with the facilities of getting an event arranged right from your home. It will provide with the means to organize an even from the scratch to completion.

Features:

1. Register

- 1.1** User will have to register in order to plan his event with the said team.

2. Login

- 2.1** Management Login: The Event Management Team will be able to access the said details for its client's event and get to know client's specifications for the event.
- 2.2** User Login: User will be able to Login and design his event.

3. Reservation

- 3.1** There will be free templates and samples so as to provide user with variety of choices and ease.

4. Invites

- 4.1** Client will be able to send invite mails right from there on the spot.

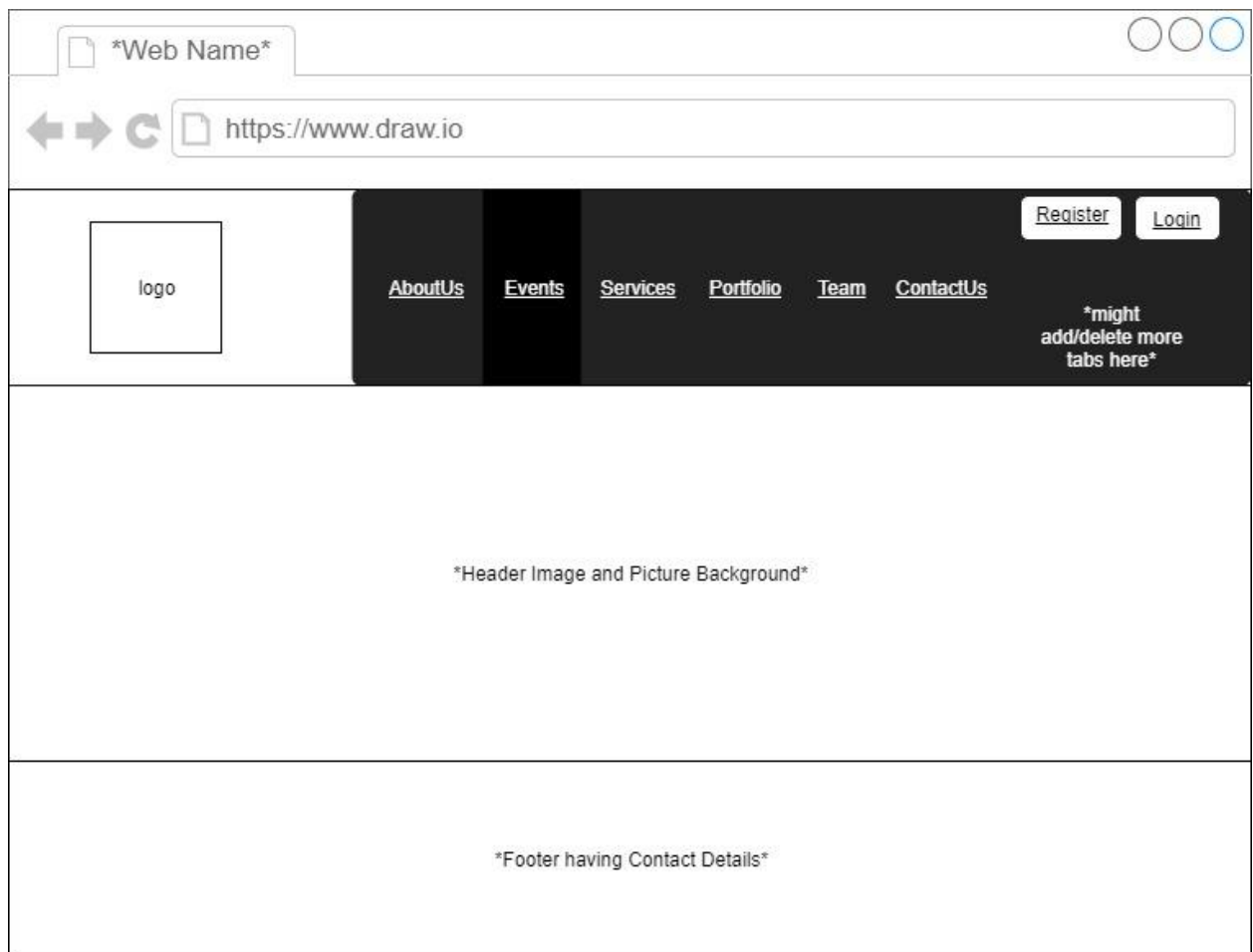
5. Feedback & Contact

- 5.1** User will be able to give feedback and get in contact with team.

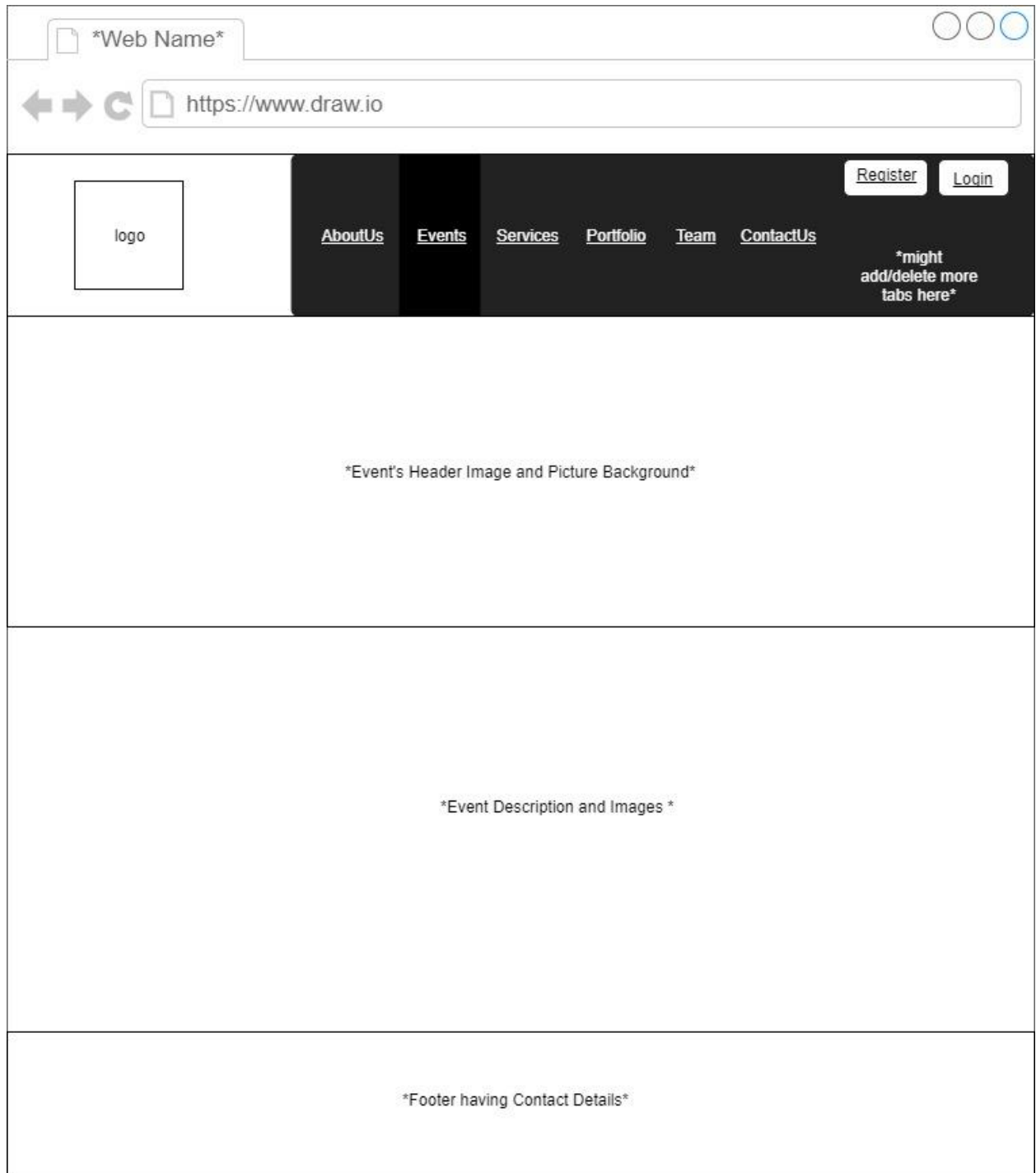
Estimated Time: 2 – 2.5 Months

“Mockups”

About Us



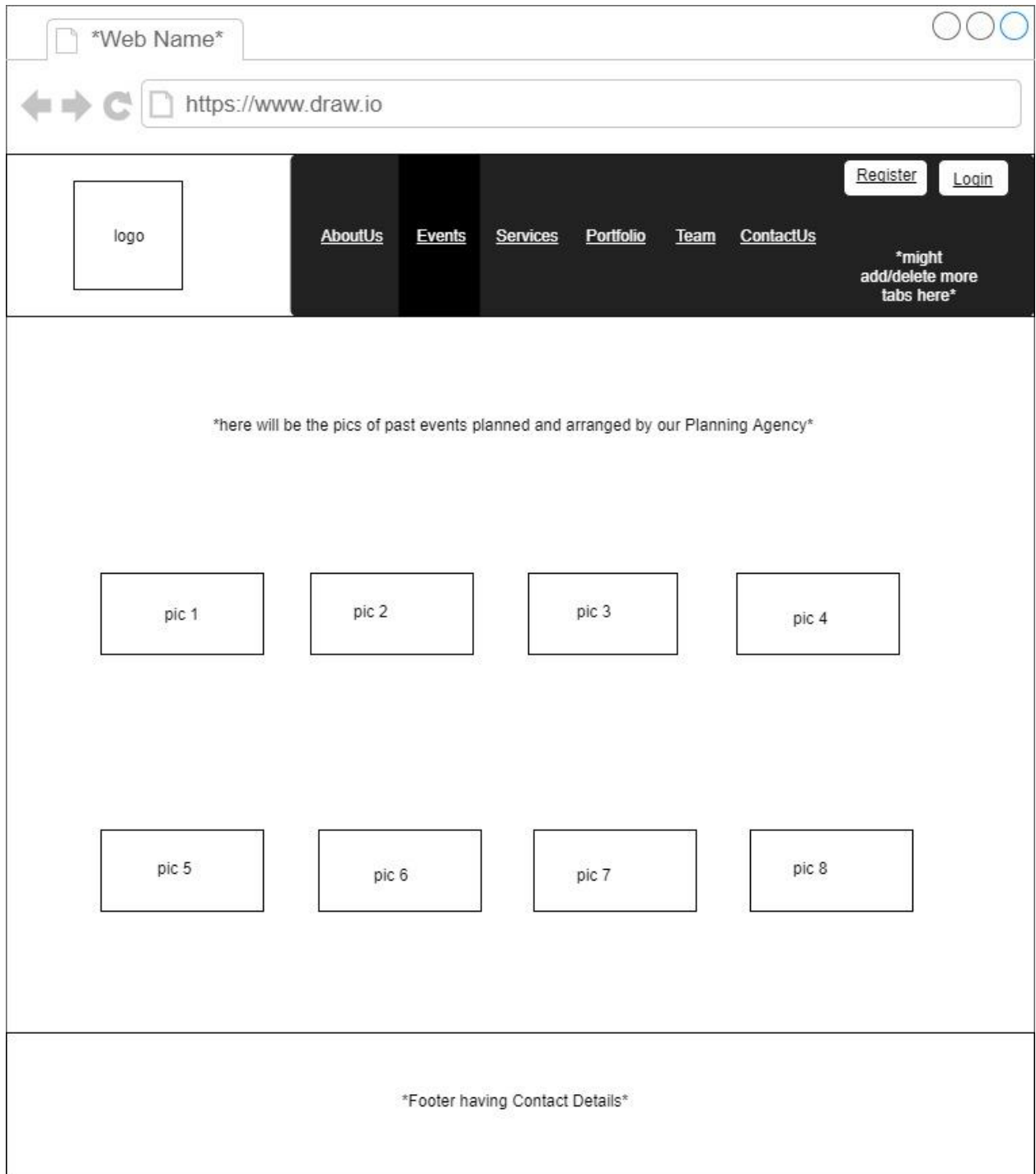
Events



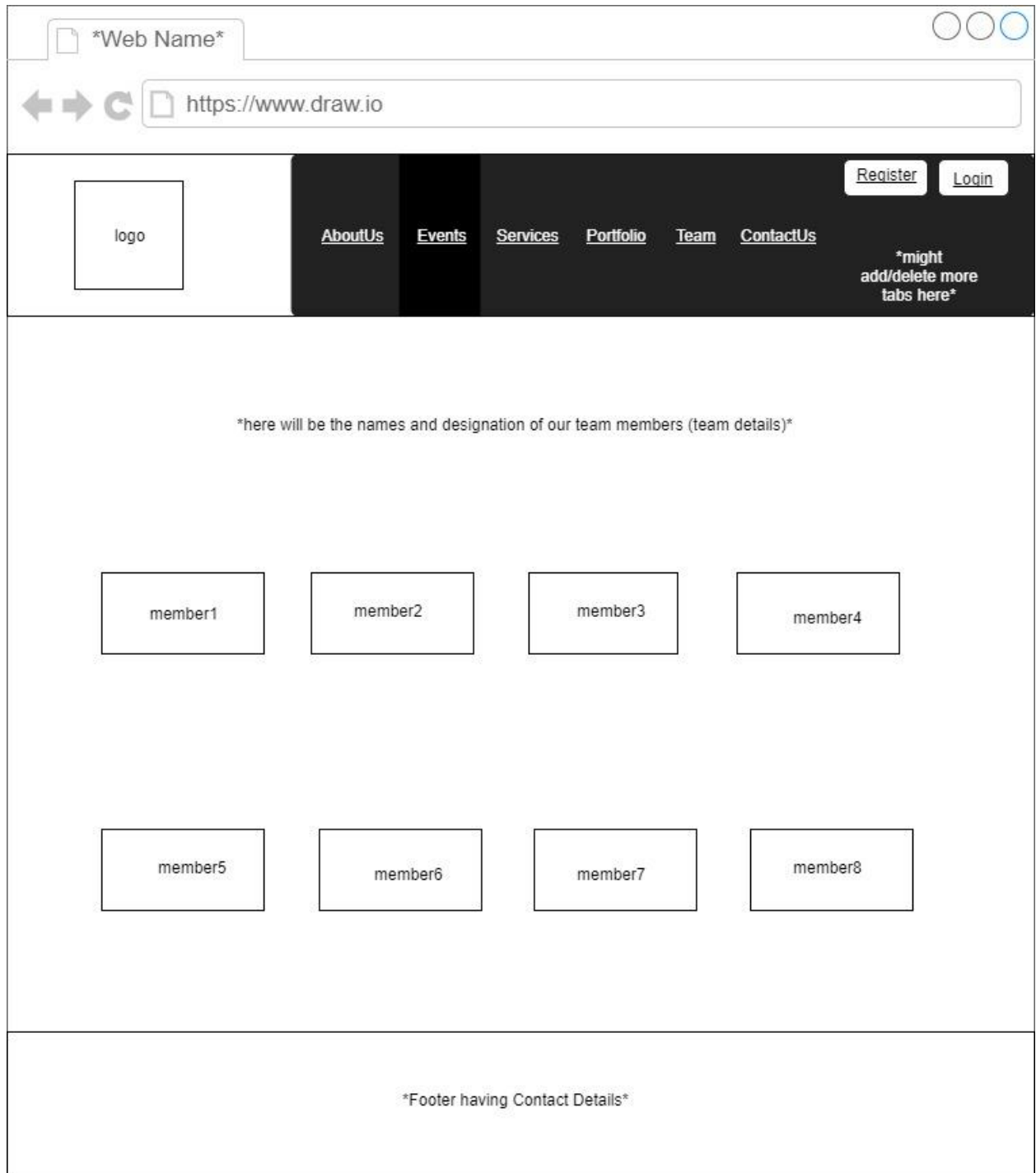
Services (Birthday Parties, Weddings, Formal Parties)

The image is a wireframe diagram of a website layout, divided into three main horizontal sections. The top section is a dark header bar. On the left, there is a placeholder for a logo labeled "logo". To the right of the logo are navigation links: [AboutUs](#), [Events](#), [Services](#), [Portfolio](#), [Team](#), and [ContactUs](#). Further right are two buttons: "Register" and "Login". Below the navigation links, there is a note: "*might add/delete more tabs here*". The middle section is a large white area, intended for the event header image and background, labeled "*Event's Header Image and Picture Background*". The bottom section is another large white area, intended for service details, labeled "*Service we provide in accordance to the type of Event Also the sample decors will be provided in options *". The footer section is a white area at the bottom, labeled "*Footer having Contact Details*".

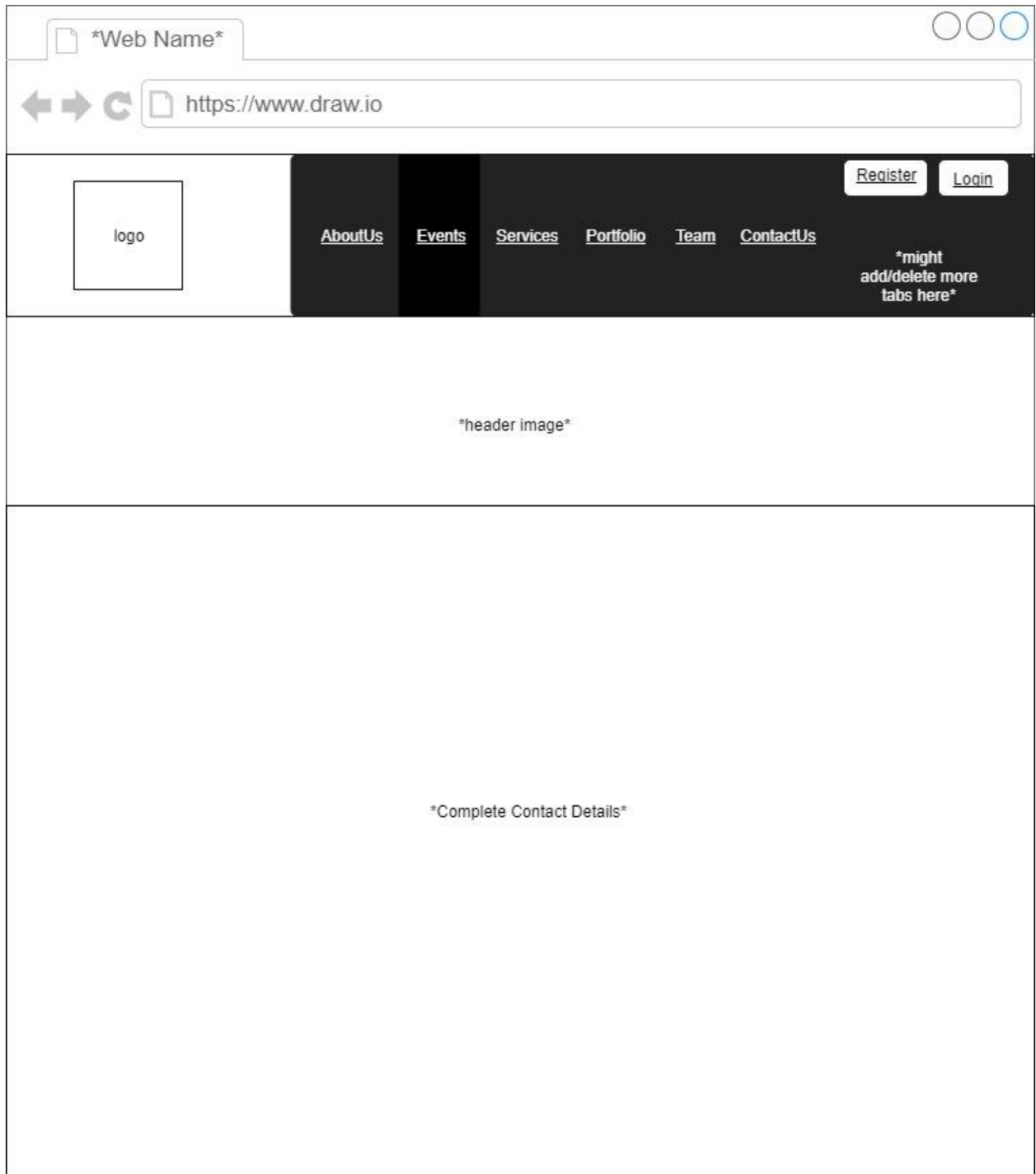
Portfolio



Team



Contact Page



Register Page

Web Name

← → ↻

https://www.draw.io

logo

[AboutUs](#)

[Events](#)

[Services](#)

[Portfolio](#)

[Team](#)

[ContactUs](#)

[Register](#)

[Login](#)

*might
add/delete more
tabs here*

Register

SignUp Form

[Proceed](#)

Footer having Contact Details

Login Page

Web Name

← → ↻

https://www.draw.io

logo

[AboutUs](#)[Events](#)[Services](#)[Portfolio](#)[Team](#)[ContactUs](#)

[Register](#)[Login](#)

*might
add/delete more
tabs here*

Reservation Itinerary

Reservation details

Login

Footer having Contact Details

Reservation Page

Web Name

← → ↻

https://www.draw.io

logo

[AboutUs](#)

[Events](#)

[Services](#)

[Portfolio](#)

[Team](#)

[ContactUs](#)

[Register](#)

[Login](#)

*might
add/delete more
tabs here*

Reservation Itinerary

Reservation details

[Submit](#)

[Cancel](#)

Footer having Contact Details

“Project Description”



The Evento – Event Planners

You Dream it, We bring it to Reality

This is the web application, a platform where users can plan their events (of any kind) online. Through this, User can select the type of event and design accordingly. It eases users in deciding the design and other important decisions for their event. It provides the ease of planning a whole event right from your home.

The Evento is made to facilitate the user in matter of organizing events in the best possible way. It consists of events designs, ideas, from which user can conduct their own event successfully. It also provides the services from decorations to catering and all. In short, it is a whole package for our customers in the matter of organizing an event easily sitting from their homes.

“API Specification Doc”

(Event Planner App)

Version	Date	Author	Description
1.0	10-Nov-2017	Amna Latif & Mah-E-Falak	Initial draft

API Index

1. User_login

Request

Response

2. Admin_login

Request

Response

3. get events booking

Request

Response

4. get contact

Request

Response

Conventions

Status Codes

Methods

1. User login

Authenticate the user with the system and obtain the auth_token

Request

Method	URL
POST	api/login/

Type	Params	Values
HEAD	User_api_key	string
POST	username	string
POST	password	string

api_key

user_api_key must be sent with all client requests. The user_api_key helps the server to validate the request source.

Response

Status	Response
200	<pre>{ "auth_key": <auth_key> }</pre> <p>auth_key(string) - all further API calls must have this key in header</p>

403	<code>{"error": "API key is missing."}</code>
400	<code>{"error": "Please provide username."}</code>
400	<code>{"error": "Please provide password."}</code>
401	<code>{"error": "Invalid API key."}</code>
401	<code>{"error": "Incorrect username or password."}</code>
500	<code>{"error": "Something went wrong. Please try again later."}</code>

2. Admin login

Authenticate the admin with the system and obtain the auth_token

Request

Method	URL
POST	<code>api/admin-login/</code>

Type	Params	Values
HEAD	admin_api_key	string
POST	admin_name	string
POST	password	string

api_key

`admin_api_key` must be sent with all client requests. The admin_api_key helps the server to validate the request source.

Response

Status	Response
200	<pre>{ "auth_key": <auth_key> }</pre> <p>auth_key(string) - all further API calls must have this key in header</p>
403	<pre>{"error": "API key is missing."}</pre>
400	<pre>{"error": "Please provide username."}</pre>
400	<pre>{"error": "Please provide password."}</pre>
401	<pre>{"error": "Invalid API key."}</pre>
401	<pre>{"error": "Incorrect username or password."}</pre>
500	<pre>{"error": "Something went wrong. Please try again later."}</pre>

3. get events booking

Get new event bookings

Request

Method	URL
POST	api/ booking

Type	Params	Values
HEAD	auth_key	String
POST	booking_id	number

auth_key

The `auth_key` that was given in response to `/api/login`

booking_id

The current version of event planner system database. Each time when new booking are reserved from the client through the web service, the internal booking_id is incremented.

Response

Status	Response
200	<p>Response will be an object containing the</p> <pre>{ "user_id": 10, "description": "Theme Birthday Party", "day": "Monday", "date": "23-01-2017", "venue": "Karachi", "decoration": "Car Theme", "catering": "Complete Dinner Meal" }, }</pre>

	<p>An example response is:-</p> <pre> { "event_booking": [{ "user_id": 10, "description": "Theme Birthday Party", "day": "Monday", "date": "23-01-2017", "venue": "Karachi", "decoration": "Car Theme", "catering": "Complete Dinner Meal" },] "event_id": "4" } </pre>
400	{"error": "Please specify event booking details."}
400	{"error": "Invalid event booking."}
401	{"error": "Invalid API key."}
500	{"error": "Something went wrong. Please try again later."}

4. get Contact Us

Get more information on a particular recipe

Request

Method	URL
GET	api/contact

Type	Params	Values
HEAD	auth_key	String
POST	email	String

auth_key

The `auth_key` that was given in response to `/api/login`

email

The `email` that was given in response to `/api/login` is compulsory for contact an admin about queries

Response

Status	Response
200	<p>An array containing the Contact Us Info</p> <pre>{ "emailid": "abc@gmail.com", "message": "Hi thanks for arranging such amazingly decorated event.", "phonenumber": "0000-0000000",</pre>

	<pre> "name": "abc", "feedback": "good" }, } </pre> <p>An example response is:-</p> <pre> { "contactus": [{ "emailid": "abc@gmail.com", "message": "Hi thanks for arranging such amazingly decorated event.", "phonenumber": "0000-00000000", "name": "abc", "feedback": "good" },] } </pre>
400	{"error": "Please provide email_id."}
400	{"error": "Invalidemail_id."}
401	{"error": "InvalidAuth key."}
500	{"error": "Something went wrong. Please try again later."}

Glossary

Conventions

- **Client** - Client application.
- **Status** - HTTP status code of response.
- All the possible responses are listed under 'Responses' for each method. Only one of them is issued per request server.
- All response are in JSON format.
- All request parameters are mandatory unless explicitly marked as [optional]
- The type of values accepted for a *request* parameter are shown the the values column like this [**10**|<any number>].The | symbol means *OR*. If the parameter is [optional], the default value is shown in blue bold text, as **10** is written in [**10**|<any number>].

Status Codes

All status codes are standard HTTP status codes. The below ones are used in this API.

2XX - Success of some kind

4XX - Error occurred in client's part

5XX - Error occurred in server's part

Status Code	Description
200	OK
201	Created
202	Accepted (Request accepted, and queued for execution)
400	Bad request
401	Authentication failure
403	Forbidden
404	Resource not found

405	Method Not Allowed
409	Conflict
412	Precondition Failed
413	Request Entity Too Large
500	Internal Server Error
501	Not Implemented
503	Service Unavailable

“README.MD”

EventPlanner

This project was generated with [Angular CLI](<https://github.com/angular/angular-cli>) version 1.5.0.

Development server

Run `ng serve` for a dev server. Navigate to `http://localhost:4200/`. The app will automatically reload if you change any of the source files.

Code scaffolding

Run `ng generate component component-name` to generate a new component. You can also use `ng generate directive | pipe | service | class | guard | interface | enum | module`.

Build

Run `ng build` to build the project. The build artifacts will be stored in the `dist/` directory. Use the `-prod` flag for a production build.

Running unit tests

Run `ng test` to execute the unit tests via [Karma](<https://karma-runner.github.io>).

Running end-to-end tests

Run `ng e2e` to execute the end-to-end tests via [Protractor](<http://www.protractortest.org/>).

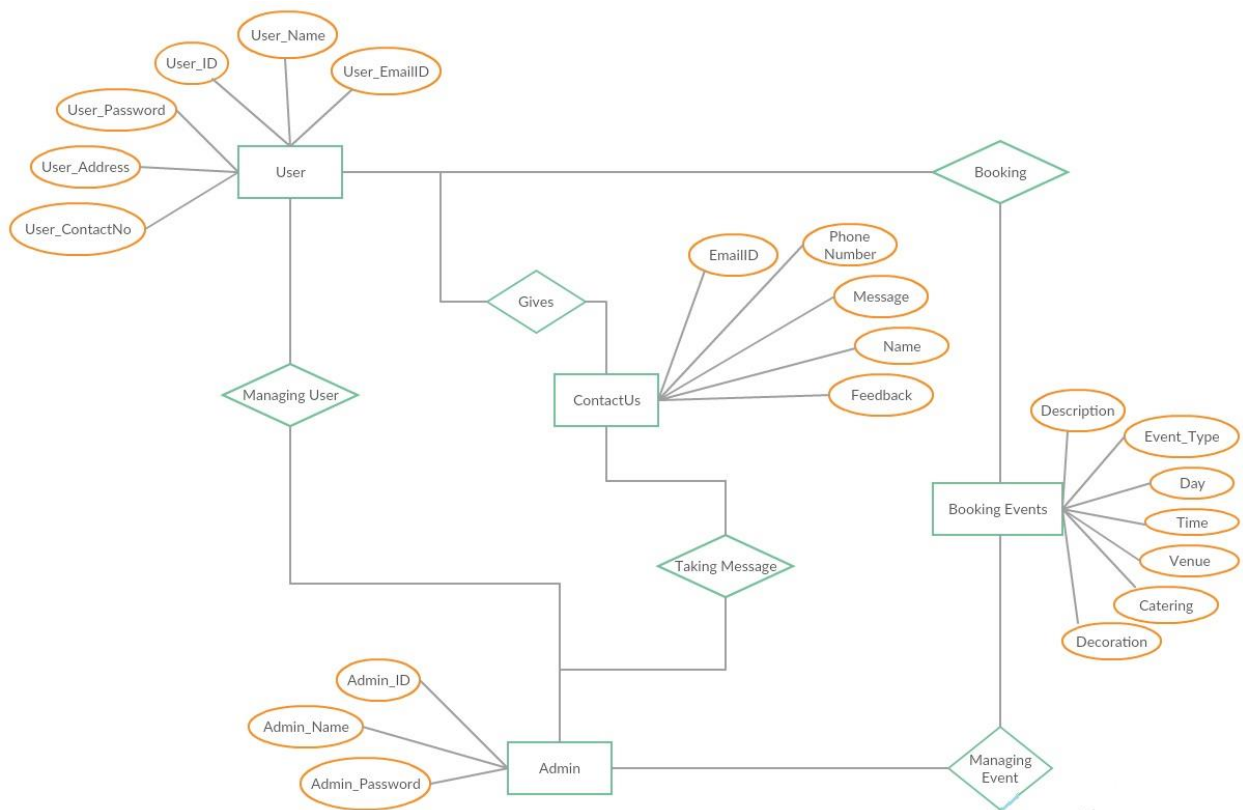
Further help

To get more help on the Angular CLI use `ng help` or go check out the [Angular CLI README](<https://github.com/angular/angular-cli/blob/master/README.md>).

“Project Complete Code Link”

<https://drive.google.com/open?id=1yzidIWknWZkKo-vi701ZiwarrgxJOwz7>

“ERD”



Google Drive Link For Complete Project Documentation Files

<https://drive.google.com/open?id=0B9y9pjpNmYsjYnd2akctZXV5alk>

||-----The End-----||