The Outdoor Art Group

Business Functionality

# Router

The Outdoor Art website will include 2 main functions: one is a public site and the other is main site.

**Front-End side will consist of these URLs:**

* Home page’s URL will be **project.app/home**. This page shows the index page which contains general information about the Outdoor Artists services.
* About page’s URL **project.app/about**. This page shows the history, vision, and mission of the company.
* Services page’s URL **project.app/services**. This page shows all services offered by the Outdoor Artists and
* Services page thumbnail’s URL: **project.app/services/thumbnail\_gallery.** This page contains a general gallery which covers all 5 services.
* Client page’s URL **project.app/services/clients.** This page specifically shows names and descriptions of the potential clients such as Estates, Commercial, and Government.
* Specific client’s URL will be **project.app/services/clients/name**. This page offers details about the custom package for each potential clients of Outdoor Artists.
* Galleries URL will be **project.app/services/clients/name/gallery**. This page summarizes 5 services with only one picture for each service.
* There are 5 pages (MTE.html, stonebrick.html, landscaping.html, fertilization.html, and maintenance.html) under each client. The link should be **project.app/services/clients/name/services/name** and it should display an individual gallery’s unique service description, features, and pictures for each client. Therefore, a total of **15 different URLs** for galleries and information will be created:
* project.app/services/client/estates/services/MTE
* project.app/services/client/estates/services/stonebrick
* project.app/services/client/estates/services/landscapingdesign
* project.app/services/client/estates/services/fertilization
* project.app/services/client/estates/services/maintenance
* project.app/services/client/commercial/services/MTE
* project.app/services/client/commercial/services/stonebrick
* project.app/services/client/commercial/services/landscapingdesign
* project.app/services/client/commercial/services/fertilization
* project.app/services/client/commercial/services/maintenance
* project.app/services/client/government/services/MTE
* project.app/services/client/government/services/stonebrick
* project.app/services/client/government/services/landscapingdesign
* project.app/services/client/government/services/fertilization
* project.app/services/client/government/services/maintenance
* Contact’s URL will be **project.app/contact.** This page shows a contact form which allows new or existing clients fill out first name, last name, phone number, email and comments the services they are interested in.

**Back-End side will consist of 3 URLs:**

* Admin Login URL will be **project.app/admin-login**. The page asks for admin’s username and password to log in.
* Admin Logout URL will be **project.app/admin-logout**. This page shows message notifying admin user successfully log out.
* Admin Edit URL will be **project.app/admin-edit**. This page shows the interface where admin can add, edit and modify their posts within the Outdoor Arts website.

# Controllers

## • How are the URLs going to be grouped?

Controllers will analyze what it receives from the router (e.g. services from the database) and it also collect information from query string (e.g. question mark id). Controllers can group incoming requests from clients’ browsers into a class and perform matching activity.

1. **Project.app/home** will be grouped by **HomeController** class which returns view of the home page and consists of the index function.
2. **Project.app/about** will be grouped by **HomeController** class which returns view of the about page and consists of the about function.
3. **Project.app/services** will be grouped by **ServicesController** class which displays a general gallery thumbnail of all services offered by the Outdoor Artists and its potential clients (Estates, Commercial and Government).
4. **Project.app/services/stonebrick** will be grouped by **ServicesController** class which returns names and description about stone and brick work functions, etc. Each function will retrieve data from database requesting gallery and information if necessary.
5. **Project.app/services/landscapedesign** will be grouped by **ServicesController** class which returns names and description about landscaping and design functions. Each function will retrieve data from database requesting gallery and information if necessary.
6. **Project.app/services/fertilization** will be grouped by **ServicesController** class which returns names and description about lawn\_fertilization function. The function will retrieve data from database requesting gallery and information if necessary.
7. **Project.app/services/maintenance** will be grouped by **ServicesController** class which returns names and description about maintenance function. The function will retrieve data from database requesting gallery and information if necessary.
8. **Project.app/services/estates** will be grouped by **ClientEstatesController** class which returns info about a project proposal for the estates client.

* **Project.app/services/estates/gallery** will be grouped by **ClientEstatesController** class which displays all five services for a defined estates client.
  1. **Project.app/services/estates/MTE** will be grouped by **ClientEstatesController** class which returns detailed info and gallery about the mowing, trimming, and edging services for estates client.
  2. **Project.app/services/estates/stonebrick** will be grouped by **ClientEstatesController** class which returns detailed info and gallery about the stone and brick services for estates client.
  3. **Project.app/services/estates/landscapingdesign** will be grouped by **ClientEstatesController** class which returns detailed info and galleries about the landscaping design service for estates client.
  4. **Project.app/services/estates/fertilization** will be grouped by **ClientEstatesController** class which returns detailed info and gallery about the lawn fertilization service for estates client.
  5. **Project.app/services/estates/maintenance** will be grouped by **ClientEstatesController** class which returns detailed info and gallery about the monthly maintenance service for estates client.

1. **Project.app/services/commercial** will be grouped by **commercialController** class which returns info about a project proposal for the commercial client.

* **Project.app/services/commercial/gallery** will be grouped by **commercialController** class which displays all five galleries for a defined commercial client.
  1. **Project.app/services/commercial/MTE** will be grouped by **commercialController** class which returns detailed info and gallery about the mowing, trimming, and edging services for commercial client.
  2. **Project.app/services/commercial/stonebrick** will be grouped by **commercialController** class which returns detailed info and gallery about the stone and brick services for commercial client.
  3. **Project.app/services/commercial/landscapingdesign** will be grouped by **commercialController** class which returns detailed info and gallery about the landscaping design service for commercial client.
  4. **Project.app/services/commercial/fertilization** will be grouped by **commercialController** class which returns detailed info and gallery about the lawn fertilization service for commercial client.
  5. **Project.app/services/commercial/maintenance** will be grouped by **commercialController** class which returns detailed info and gallery about the monthly maintenance service for commercial client.

1. **Project.app/services/government** will be grouped by **governmentController** class which returns info about a project proposal for the government client.

* **Project.app/services/government/gallery** will be grouped by **governmentController** class which displays all five galleries for a defined government client.
  1. **Project.app/services/government/MTE** will be grouped by **governmentController** class which returns detailed info and gallery about the mowing, trimming, and edging services for government client.
  2. **Project.app/services/government/stonebrick** will be grouped by **governmentController** class which returns detailed info and gallery about the stone and brick services for government client.
  3. **Project.app/services/government/landscapingdesign** will be grouped by **governmentController** class which returns detailed info and gallery about the landscaping design service for government client.
  4. **Project.app/services/government/fertilization** will be grouped by **governmentController** class which returns detailed info and gallery about the lawn fertilization service for government client.
  5. **Project.app/services/government/maintenance** will be grouped by **governmentController** class which returns detailed info and gallery about the monthly maintenance service for government client.

1. **Project.app/contact** will be grouped by **contactController** class which returns view of the contact page and consists of contact functions such as post, comments, etc.
2. **Project.app/admin-login** will be grouped by **adminLoginController** class which return a view of admin’s login page which prompts for username and password.
3. **Project.app/admin-edit** will be grouped by **adminEditController** class which return a view of admin’s edit page which allow admin users to edit the page.
4. **Project.app/admin-logout** will be grouped by **adminLogoutController** class which return a view of admin’s logout page which notify admin users log out successfully.

## • How will the response be handled? What will be returned?

* There are 2 types of requests: one is from the public side and one is from the admin side. The controller will request data from database through model if the request related to the database query. For example, the request could be the maintenance service for the government client.
* After receiving data from database, the data will travel back to the model and controller before transmitting to the view. Finally, the browser will display the data on the browser.
* The service features’ information, galleries, and descriptions as well as clients custom packages will be returned.

## • What data do you need?

The data that we need to retrieve from database is:

* For galleries and services: a list of services, a list of galleries and a thumbnail gallery (Mowing, Trimming, Edging gallery; Stone and Brick gallery; Landscaping and Design gallery; Lawn Fertilization gallery; and Monthly Maintenance gallery.)
* For clients: a list of clients (Estates, Commercial, and Government) and a list of custom package of a specific client.

• What views are needed?  
The view would be:

* For home page: ‘index’
* For about page: ‘about’
* For services page: ‘services\_list’
* For client list: ‘services/clients\_list’ (services is a folder)
* For a specific client:

Estate: (there will be a folder called services/estate)

‘services/estates/galleries’,

‘services/estates/MTE’,

‘services/estates/stonebrick’,

‘services/estates/landscapingdesign’,

‘services/estates/fertilization’,

‘services/estates/maintenance’

Commercial: (there will be a folder called services/commercial)

‘services/commercial/galleries’,

‘services/commercial/MTE’,

‘services/commercial/stonebrick’,

‘services/commercial/landscapingdesign’,

‘services/commercial/fertilization’,

‘services/commercial/maintenance’

Government: (there will be a folder called services/government)

‘services/government/MTE’,

‘services/government/stonebrick’,

‘services/government/landscapingdesign’,

‘services/government/fertilization’,

‘services/government/maintenance’

* For galleries page: ‘services/galleries\_list’ and ‘services/thumbnail\_gallery’
* For contact page: ‘contact’
* For admin login page: ‘admin-login’
* For admin edit page: ‘admin-edit’
* For admin logout page: ‘admin-logout’