

## Workout Tracker

[View All](#) [Create](#) [Category](#) [Track](#)

Morning Run for 5 min

Edit

Delete

Start

End

Boxing practice for 10 min

Edit

Delete

Chest and tricep workout

Edit

Delete

Evening walk for 30 min

Edit

Delete

# SINGLE PAGE APPLICATION WORKOUT TRACKER

MEAN FSD

### ABSTRACT

Personal Workout Tracker single page app; which is used to manage personal workout and fitness plans.

### Developer

MEAN Full Stack Developer

## Contents

Important Instructions	1
Business-Requirement: An Overview	2
Business-Requirement: Summary	2
Technical Spec – Solution Development Environment	3
Front End Layer	3
Middle Tier Layer	3
Database & Integration Layer	3
Ancillary Layer	3
Deployment & Infrastructure	4
Editors	4
Wireframes	5
Wireframe: Start Workout Screen	5
Wireframe: Track Workout screen	12
Software Requirements	13
Database Structure	14
Architecture	14
Architecture Diagram for MEAN Full Stack	16
Important Instructions	17
Assessment Deliverables	17

## Important Instructions

1. Adhere to the design specifications mentioned in the case study.
2. Please make sure that your code does not have any compilation errors while submitting your case study solution.

3. The final solution should be a zipped code having solution. Solution code will be used to perform Static code evaluation.
4. Implement the code using best design standards.
5. Use Internationalization for all the labels and messages in Rest API Development.
6. Do not use System.out statements or console.log for logging in Rest API and FrontEnd respectively. Use appropriate logging methods for logging statements/variable/return values.
7. If you are using Node and Express to develop Rest API, then use Grunt/Gulp/NPM to build/minify the project and create application for deployment.
8. Write web service which takes input and return required details from database.
9. Use JSON format to transfer the results.

## Business-Requirement: An Overview

Before beginning work on any project, it's usually a good idea to know what you're building. Below is a basic list of things we want our users to be able to do:

The client would like to develop an independent application; Personal Workout Tracker single page app; which is used to manage personal workout and fitness plans.

Workout Tracker:

1. This web application allows user to enter in and save the workout plans.
2. Each new workout entered by user will capture calories burnt per min along with
3. It will allow you to start the workout and end the workout whenever the user does the workout and ends the workout respectively.
4. It will present graphs that show your progress over time.
5. It will provide report on number of calories burnt.
6. It will provide report on number of minutes the workout was done on each day, each week and each month and also show the graph for the same.

## Business-Requirement: Summary

Workout Tracker

1. Add/Update/Delete/View Workout

2. Add/Update/Delete/View Category
3. Start Workout
4. End Workout

## Technical Spec – Solution Development Environment

### Front End Layer

Framework(s)/SDK/Libraries	Version
Bootstrap	3.0 or above
CSS	3
HTML	5
JavaScript	1.8 or above
JQuery	1.3
React	15.x.x

### Middle Tier Layer

Technology	Framework(s)/SDK/Libraries	Version
MEAN Stack	Node.JS	6.x (Latest: 6.11.4)
	Express.JS	4.16.1

### Database & Integration Layer

Technology	Framework(s)/SDK/Libraries	Version
MEAN Stack	MongoDB/MySQL	
	NoSQL	
	Mongoose.JS	4.12.1

### Ancillary Layer

Technology	Framework(s)/SDK/Libraries	Version
Source Code Management Tool	GIT	2.14.2
Build Tool/MEAN Stack	Grunt/Gulp	1.0.1/3.9.1
Dependency Management Tool/MEAN Stack	NPM	4.x
Testing Tool/MEAN Stack	Jasmine	

## Deployment & Infrastructure

Technology	Framework(s)/SDK/Libraries	Version
Docker	-	
Apache HTTP (XAMPP)	-	
Node	-	
Dependency Management Tool	NPM	

## Editors

Name	Version
Adobe Brackets	
Sublime Text	
Atom Editor	

## Wireframes

### Wireframe: Start Workout Screen

# Workout Tracker

[View All](#) [Create](#) [Category](#) [Track](#)

Morning Run for 5 min

EditDeleteStartEnd

Boxing practice for 10 min

EditDeleteStartEnd

Chest and tricep workout for 10 min

EditDeleteStartEnd

Evening walk for 30 min

EditDeleteStartEnd

It has a functionality of autofilter. It will filter the search results based on what user types in the search box. It is implemented using angular-js filter.

Remain disabled by default. It will be enabled only after the particular workout is started by clicking on start button

1. Displays the list of workout ideas entered by user.
2. User can edit, delete the workout
3. User can start the workout when user wants to start the workout. It will get disabled after the workout is started.

4. User can end the workout after user is done with the workout. It is disabled by default. It will be enabled only if workout is started.
5. The screen has a search bar with auto-filter enabled i.e. as user types the workout; he/she wants to search the results will get auto filtered below.
6. Navigation in the website happens as a SPA (Single Page Application)

#### Wireframe: Add Workout Screen

Workout Tracker

[View All](#) [Create](#) [Category](#) [Track](#)

Add Workout

Title:

Note:

Calories Burnt Per Min

Category:

Add Workout

Floating Point Numbers only .

- Clicking on + will add by 0.1
- Clicking on – will decrease by 0.1

When the page loads, 'Add Workout' is disabled. It gets enabled only after the Title, Calories Burnt Per Min and Category is selected.

1. This screen allows you to add the workout.
2. You can also manage the category to which workout belongs in this screen.

3. You will need to input the calories burnt per min if the user does this workout.
4. Ref: <https://whatscookingamerica.net/Information/CalorieBurnChart.htm>

Wireframe: [Add Workout Screen](#)

Workout Tracker

[View All](#) [Create](#) [Category](#) [Track](#)

Add Category

Search

+

Jogging Edit Delete

Sprint Edit Delete

Slow Walk Edit Delete

Cardio Edit Delete

Aerobics Edit Delete

Yoga Edit Delete

Proper validation should be in place to prevent user from entering blank category

1. It allow user to manage categories
2. The screen has auto-search box which will allow user to filter the search results while typing in the search box.



# Workout Tracker

[View All](#) [Create](#) [Category](#) [Track](#)

Edit Workout

Title:

Note:

Category:

Select Category

▼

+

Update

Cancel

1. It allows users to edit any workout they may have entered

## Wireframe: End Workout Screen

The wireframe shows a screen titled "Workout Tracker" with a sub-header "End Workout". Below this are four input fields: "Title:", "Comment:", "End Date", and "End Time". The "Title:" field is a single-line text input. The "Comment:" field is a multi-line text area. The "End Date" and "End Time" fields are single-line text inputs. To the right of the "End Date" and "End Time" fields are two blue callout boxes. The first box, connected to the "End Date" field by a line, contains the text "Date Format. By default, populated with current date". The second box, connected to the "End Time" field by a line, contains the text "Time Format. By default, populated with current time". At the bottom of the screen are two dark gray buttons: "End" and "Cancel".

Workout Tracker

End Workout

Title:

Comment:

End Date

End Time

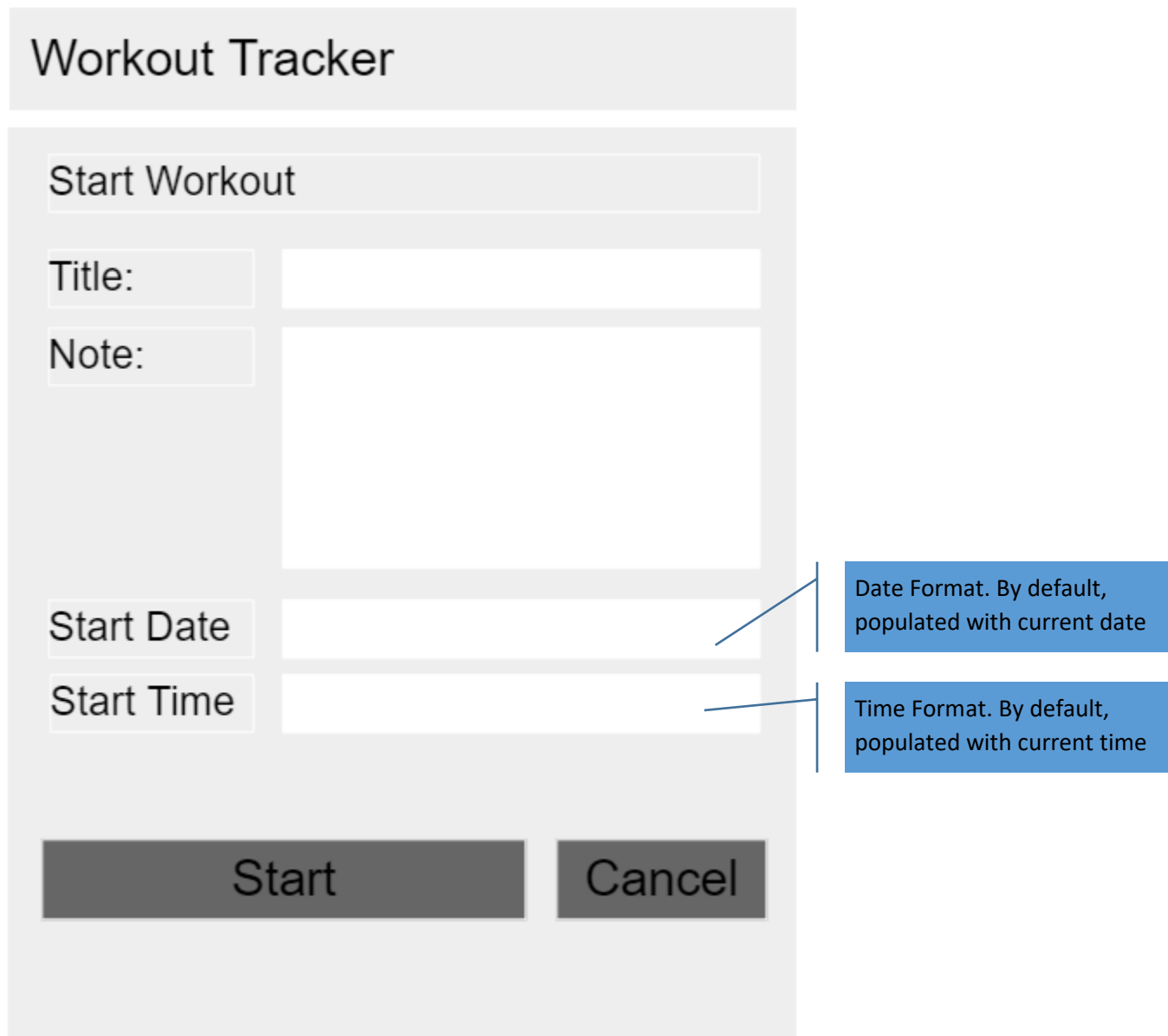
Date Format. By default, populated with current date

Time Format. By default, populated with current time

End Cancel

1. It allows users to end the workout which is already started.
2. Users can input the end date and end time.
3. By default workout title is populated. This is un-editable.
4. By default current date and current time is populated.

## Wireframe: Start Workout Screen



The wireframe shows a 'Workout Tracker' app interface. At the top is a header bar with the title 'Workout Tracker'. Below the header is a form titled 'Start Workout'. The form contains several input fields: a 'Title' field, a 'Note' field (a larger text area), a 'Start Date' field, and a 'Start Time' field. To the right of the 'Start Date' and 'Start Time' fields are two blue callout boxes. The first box, connected by a line to the 'Start Date' field, contains the text 'Date Format. By default, populated with current date'. The second box, connected by a line to the 'Start Time' field, contains the text 'Time Format. By default, populated with current time'. At the bottom of the form are two buttons: 'Start' and 'Cancel'.

Workout Tracker

Start Workout

Title:

Note:

Start Date

Start Time

Date Format. By default, populated with current date

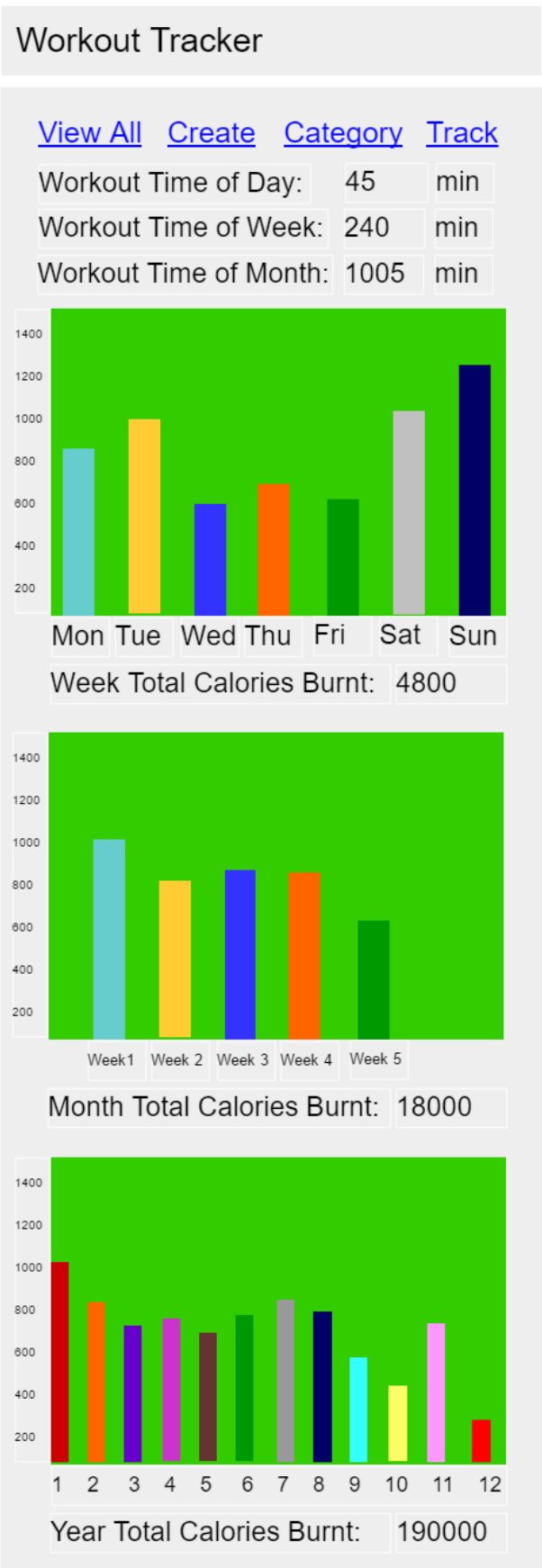
Time Format. By default, populated with current time

Start Cancel

1. It allows users to start the workout.
2. Users can input start date and start time.
3. By default workout title is populated. This is un-editable.
4. By default current date and current time is populated.



Wireframe: Track Workout screen



1. This page displays various reports:
  - a. Current day workout time. It will calculate the sum of all the workouts done on the current day.
  - b. Current week workout total time. It will calculate the sum-time of all the workouts done in the week.
  - c. Current month workout total time. It will calculate the sum-time of all the workouts done in the month.
  - d. Charts:
    - i. Week wise calorie burn chart
    - ii. Month wise calorie burn chart
    - iii. Year wise calorie burn chart

## Software Requirements

This case study assumes knowledge of programming and hands-on with below mentioned skills.

The technologies included in Full Stack are not limited to following but may consist of:

- UI Layer (HTML5, CSS3, Bootstrap, JavaScript, JQuery, AngularJS, JSP)
- Middleware Restful API (Node, Express)
- Database Persistence ( MongooseJS)
- Database layer (MongoDB, NoSQL)
- Ancillary skills (GIT, Docker, Grunt/Gulp) etc.

To complete this case study, you should be comfortable with basic single page web application concepts including REST and CRUD. The environment setup is built into virtual environment you are logged in to.

You may use angular-cli to create your template project.

Ref1: <https://cli.angular.io/>

Ref2: <https://github.com/angular/angular-cli>

## Database Structure

workout_collection		
workout_title	varchar_128	
workout_note	varchar_256	
calories_burn_per_min	float	
category_id	integer	foreign
workout_id	integer	primary

workout_active		
workout_id	integer	foreign
start_time	time	
start_date	date	
end_date	date	
end_time	time	
comment	varchar_64	
status	boolean	

workout_category		
category_id	integer	primary
category_name	varchar_64	

## Architecture

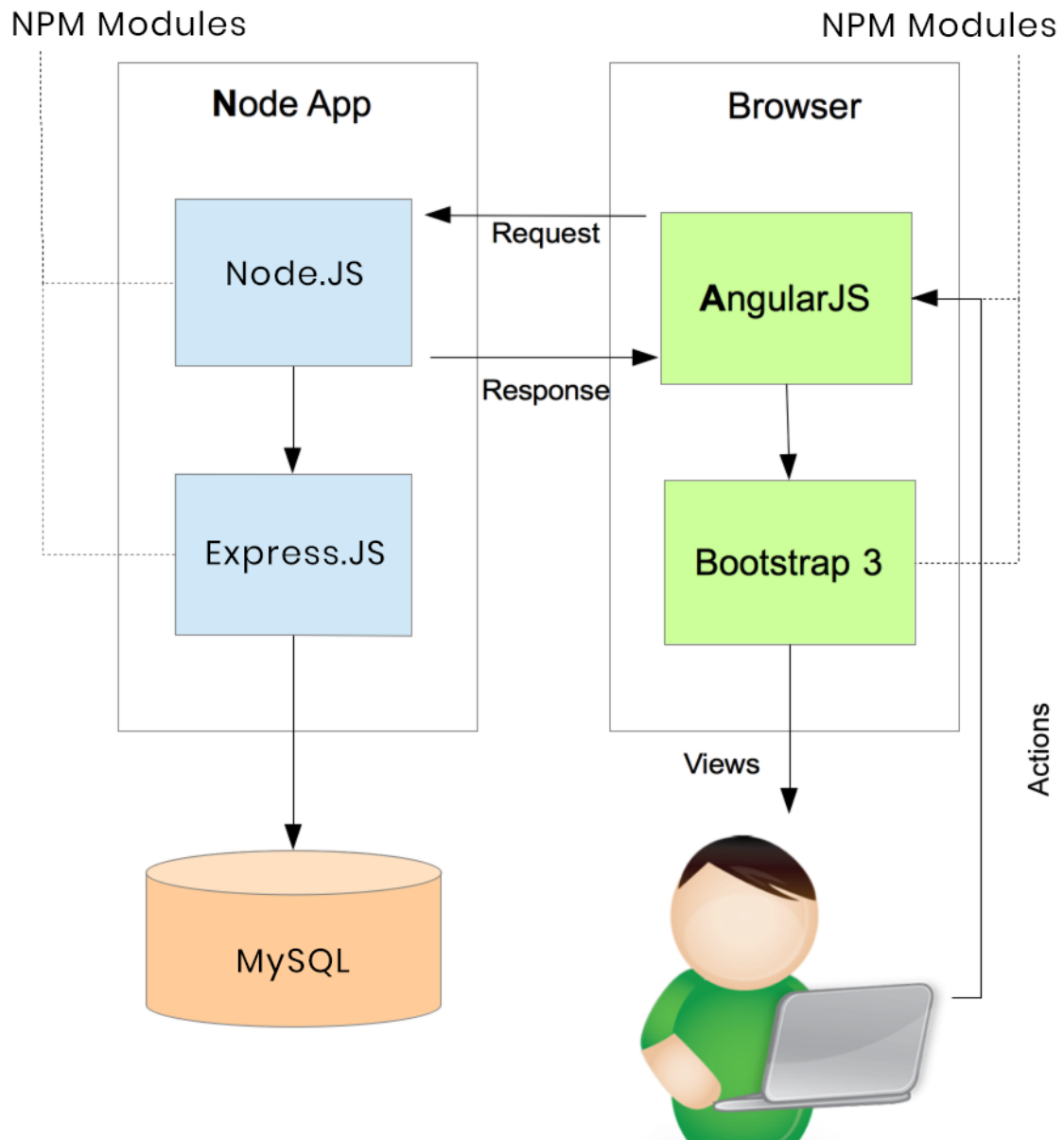
A physical architecture is an arrangement of physical elements, (system elements and physical interfaces) that provides the designed solution for a product, service, or enterprise. It is intended to

satisfy logical architecture elements and system requirements. Workout Tracker follows a three layered architecture namely presentation layer, business logic layer and data access layer.

- **Presentation Tier** is the tier in which the users interact with an application. It is a single-page-application of responsive nature. Presentation Tier will consume restful API implemented in Business Tier to display content to the user.
- **Business Tier** is mainly working as the bridge between Data Tier and Presentation Tier. All the Data passes through the Business Tier before passing to the presentation Tier. Business Tier is the sum of Business Logic Layer, Data Access Layer and Value Object and other components used to add business logic. It exposes Rest API which can be called by Presentation Tier to display content to the user. It will also send the data from Presentation tier to Data Tier using Rest API.
- **Data Tier** is basically the server which stores all the application's data. Data tier contents Database Tables, XML Files and other means of storing Application Data.



## Architecture Diagram for MEAN Full Stack



## Important Instructions

12. Adhere to the design specifications mentioned in the case study.
13. Feel free to create front-end and back-end of single page application from scratch. You are free to use angular-cli commands at command prompt to create SPA (Single Page Application) template.
14. Please make sure that your code does not have any compilation errors while submitting your case study solution.
15. The final solution should consist of three parts:
  - a. Front-end built using HTML5, CSS3, Bootstrap and Angular JS as a SPA. Controllers written in Angular JS should consume restful API coded in business layer.
  - b. Business-layer built using Node.JS/Express.JS.
  - c. Database-layer built using MongoDB.

## Assessment Deliverables

1. For Front End – zipped application
2. For Backend – zipped application
3. For SCM\* – Project Code should be present in active GIT repository
4. Few steps on how to run the solution

\*SCM – Source Code Management