# Node.js Workshop – Blog System

## Route GET / (5 Points)

Create a Node.js web server which is capable of returning an “index.html” file containing a welcome message and menu. Add all available routes from below to the menu. On this page visualize 6 articles, sorted by total views.

## Route GET /create (5 Points)

The server should be able to return a simple form for creating an article. Each article should have Title and Description. The form should have two inputs – one for the title and one for the description.

## Route POST /create (20 Points)

When the form is submitted, the server should set some unique ID to the article and save it on some “database”. Database can be a simple in memory array of objects or a JSON file. All fields should be validated and should not be empty. You can process the invalid input in whatever way you see fit (a simple error message is more than enough).

## Route GET /all (10 Points)

When this route is reached the server should return dynamically generated HTML containing list of all articles with their title and date of creation. Deleted articles should not be shown in the list. Articles should be listed sorted by their date of creation in ascending order. Each article should have a link to “/details/{id}” where “id” is the ID of the article. Links are written with <a href=”url”>My link</a>.

## Route GET /details/{id} (10 Points)

When this route is reached, details about the article with the provided ID should be showed. When this URL is open a view should be counted on the article with the provided ID. The title, the description and the total views should be displayed. A button named “DELETE” should exist on the page. When clicked, the server should mark the article as deleted.

## Add undelete option (5 Points)

Allow the article to be undeleted. Change the button text and behavior depending on the state of the article.

## Add form for comments on the article details (5 Points)

Add form to the article details allowing the user to save comments for the current article. The form should have two fields - for username and for the comment text.

## Route POST/details/{id}/comment (15 Points)

The form for comments should save the data on the above route. The comment should be saved for the article with the provided ID. Validate the comment. It should not be empty. Save the date of the comment too. Add on the “/details/{id}” page all comments made for the corresponding article.

## Add image upload for each article (20 Points)

Add an input type “file” to the “/create” form allowing for the user to save an image for each article. Save the file on a server folder named “images”. Change the file name so that there will not be any collisions if the user sends two files with the same name. Show the file on the article details page. Use <img src=”imageSrc” /> HTML tag.

## Add statistics page (5 Points)

Add GET /stats route which is available only if “My-Authorization” header with “Admin” value is provided in the request (otherwise return 404). Show all article listed (including the deleted ones), the total number of comments and total number of views.

**OTHER REQUIREMENTS**

* Use Node.js as web server
* Add initial sample data for easier testing of the application
* Follow the standard or other linter - <https://github.com/feross/standard>
* You may use whatever frameworks you like (including express.js)