Questions:

1.How to secure form submission?

Just use a proper password authentication approach, and no-one will be able to break anything unless they know the password (regardless of where the HTTP requests are coming from). Once you have reliable server-side authentication, you don't need to waste time jumping through non-robust hoops worrying about this scenario.

2.How would you optimize a website's assets/resources?

In order to optimize your website's assets you need to understand the basic optimization rules. the basics include that you should decrease download sizes and make fewer http requests. For optimizing the website's assets you should use the following techniques:

* Name the assets
* Use a content delivery network
* Host assets on different domains but reduce dns lookups
* Place assets on a cookie-free domain and split the assets among domains
* Use CSS Sprites
* Disable etags

The above methods will be useful in reducing the page load time as well.

3.What UI, Security, Performance, SEO, Maintainability or Technology considerations do you make while building a web application or site?

* **UI**: I like minimal UI which contains only what it should. I believe it results in the better user experience, as a user knows what to do intuitively.
* **Security**: I always try to make both frontend and backend secure, concerning CSRF, XSS, etc.
* **Performance**: I consider space and time complexity for the algorithms and logics I use and write.
* **SEO**: Set meta tags for search engines and consider and consider server-side rendering for SPA.
* **Maintainability**: Try to keep the source code consistent and make objects immutable. Use statically typed languages such as TypeScript. Use CI with tests and lints.
* **Technology**: I like to learn new technologies, but if the project is in production, I would consider using technologies which is well-documented and widely used.

4.What do you know about design pattern?

Design patterns can speed up the development process by providing tested, proven development paradigms. Effective software design requires considering issues that may not become visible until later in the implementation. Reusing design patterns helps to prevent subtle issues that can cause major problems and improves code readability for coders and architects familiar with the patterns.

Design patterns can speed up the development process by providing tested, proven development paradigms. Effective software design requires considering issues that may not become visible until later in the implementation. Reusing design patterns helps to prevent subtle issues that can cause major problems and improves code readability for coders and architects familiar with the patterns.

5.What are HTTP methods? List all HTTP methods that you know, and explain them.

The Hypertext Transfer Protocol (HTTP) is designed to enable communications between clients and servers.

HTTP works as a request-response protocol between a client and server.

A web browser may be the client, and an application on a computer that hosts a web site may be the server.

Two commonly used methods for a request-response between a client and server are: GET and POST.

* **GET** - Requests data from a specified resource
* **POST** - Submits data to be processed to a specified resource

6.Database: What is the difference between a View and a Table?

Data in tables may or may not be physically stored in the database. There are two types of tables, namely; object tables which use an object type to define a column and hold instances of a defined object, and a relational table which holds basic user data in a relational database.

View, on the other hand, is a query used as a table which can be linked to another table. It is a list of several records in a table format that are used for data that are queried frequently. In querying the names and addresses which are located at different tables, views can be used. It is a virtual table that is gathered from data in the database

7.What's Two Factor Authentication? How would you implement it in an existing web application?

In an ideal world, we would all be authenticating ourselves using tamper-proof hardware that implements a public-key cryptographic system. In the meantime, a simple and effective way of improving the way your users authenticate themselves is a method known as “[Two-Factor Authentication](http://www.twilio.com/solutions/two-factor-authentication)“, “Two-Factor Auth”, or just “TFA”. Two-Factor Authentication is a method where your users are required to log in with two “factors”: a password, and a code from a device that they carry on their person. That device used to be a special-purpose device, but these days that device can just as well be a mobile phone.

A great pattern that we are seeing for implementing [two-factor authentication](http://www.twilio.com/solutions/two-factor-authentication) is to use the TOTP (Time-based One-time Password Algorithm) standard for the second authentication step