1. General Information on web browser
    - Files required for displaying a web page on the browser.
    - Understanding the Document Object Model (DOM)
      (How to access and manipulate single element in the DOM using javascript)
      (Selector Performance)
    - HTTP requests (Get, Post, Put, Delete) and response.
    - Existence of CORS.
    - Overview of the Chrome Development Tools.
        - Modify HTML/CSS
        - Stepping through front end code and adding breakpoints.
        - Using the Network tab.
        - Check cookies from the Application tab.
    - Minification.
2. Styles and HTML
    - Difference between block and inline html tags.
    - Familiarity with the CSS box model and CSS units (px, em, rem, % etc)
    - Common CSS properties
        - Border, Padding, Margin, Display, Position, Float, Text
        - CSS specificity
        - Flexbox, CSS grid(Optional)
    - Common css classes from Bootstrap
        - Margin, padding css classes.
        - Bootstrap responsive grid css classes.
        - Bootstrap css classes for input form elements.

3. Javascript
    - Basics variables and functions.
    - Asynchronous Javascript and event loop.
    - Array functions.
    - Event bubbling and capturing, Event Delegation.
    - Closures
    - JSON, json.stringify and json.parse
    - Promises
    - Async Function Expression and Await and Async
    - Timing functions
    - Spread and Rest Syntax.

4. Angular
    - Difference between server rendered and client rendered web application.
    - Single page application.
    - Why Typescript ?
    - Decorators in Typescript.
    - Generics in Typescript.
    - Angular building blocks components and modules.
      (Build sample project to learn)
        - Understanding Declarations, Imports, Exports, Providers and EntryComponents arrays in a module.
        - View encapsulation in components.

        - Life-cycle hooks of a component.
    - Inter component communication @Input and @Output decorators.
    - Routing
        - Defining and navigating routes as well as child routes.
        - Accessing query parameters and angular component reuse strategy.
        - Route guards.
    - Dependency injection
        - Defining providers useExisting, useValue, and useFactory.
        - Understanding hierarchy of the injectors and lifetime of the service.
    - Built in services (HTTP interceptor service, HTTP service)
        - Get, Post, Put, Delete api calls and error handling.
        - Adding common header to all the API calls (HTTP Interceptors).
        - Chaining multiple API calls using observables (Handling observables).
    - Data-Binding (one way and two way data binding)
    - Lazy-Loading modules
    - NPM adding third party packages
    - Using angular CLI
        - To generate new angular components, modules, pipes, directives etc
        - To build an angular project for production environment. (Different build parameters)
        - To serve an angular project for development.

    - Understanding Template driven and Reactive forms. More focus should be given on Template driven forms.
    - Validating Form Input.
        - Using built in validations.
        - Adding custom validations
        - Displaying validation error messages.
    - Knowledge about ng-container, ng-template, ng-template-outlet.
    - Knowledge about @ViewChild decorator.
    - Knowledge of Renderer2 API.
    - Built in directives (NgClass, NgStyle, NgFor, NgSwitch and NgIf)
    - Knowledge of all the files and folders (package.json, angular.json, tsconfig.json etc) of an angular project.
    - Knowledge of linting via tslint or eslint and also make use of prettier.

5. **Coding Standards**

6. Test project at the end of the training.

DAY-1

- Add/Update Account
    - Account Type,
    - Interest Rate,
    - Account Name,
    - Account Number(Autogenerated and uneditable)
    - Date Of Creation(Autogenerated and uneditable)

    - Account Balance(uneditable)
    - Miscellaneous fields can be added as per the requirement.
- Account Master should have account add/edit/delete function
    - Table/Grid mandatory.
    - Each account can be edited and deleted from the grid.

DAY-2

- Create Transaction Page
    - Withdraw / Add Money (Selection should be editable)
    - Account number and Account Name should be pre-populated and uneditable.
- The Transaction button should be present in the Account Master next to the Edit/Delete action buttons.

Resources:

- https://css-tricks.com/ (Reference for all CSS properties)
- https://flexboxfroggy.com/ (Interactive way to learn flexbox)
- https://cssgridgarden.com/ (Interactive way to learn CSS Grid)
- Angular:Getting Started
- Angular Fundamentals (Course to start with)
- Angular Routing (Lazy Loading, More on Routing Guards)
- Angular Reactive Forms
- Angular Component Communication

Useful VS Code Extensions

- Angular Language Service
- Angular Schematics
- Auto Import
- Code Spell Checker
- Move TS
- Path Intellisense
- TS Lint
- ES Lint
- Prettier