CAP - 785 (Web Optimization)

CA - 3

SUBMITTED By :-

Name :Rajanikant Upadhyay

Registration No:12315860

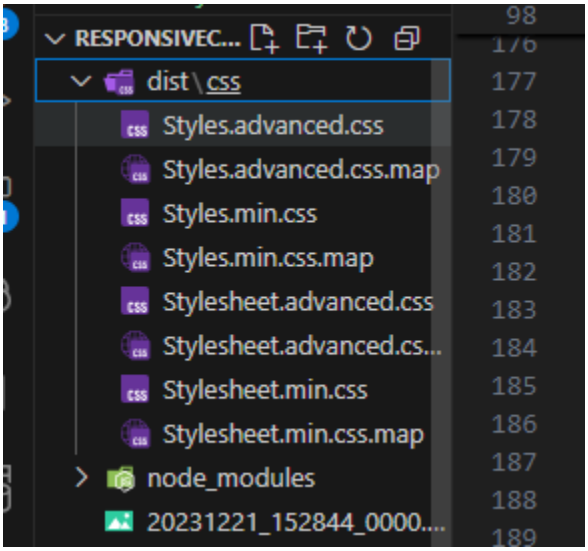
SUBMITTED To : -

Name :Dr. Wasiur Rhmann

Q1. Use website of CA2 and apply Gulp for CSS and JavaScript optimization

Gulp install command :- **npm install --save-dev gulp gulp-postcss gulp-clean-css autoprefixer cssnano gulp-concat gulp-sourcemaps gulp-rename postcss-sorting postcss-combine-media-query postcss-combine-duplicated-selectors**

OUTPUT:- It create dist file



gulpfile.js

```
const gulp = require('gulp');
const postcss = require('gulp-postcss');
const autoprefixer = require('autoprefixer');
const cssnano = require('cssnano');
const cleanCSS = require('gulp-clean-css');
const concat = require('gulp-concat');
const sourcemaps = require('gulp-sourcemaps');
const rename = require('gulp-rename');

// **PostCSS Plugins**
const sorting = require('postcss-sorting');
const combineMediaQueries = require('postcss-combine-media-query');
const combineSelectors = require('postcss-combine-duplicated-selectors');

// **Paths**
const paths = {
  css: {
    src: '*.css',  // Input CSS files
    dest: 'dist/css/'         // Output directory
  }
};

// **1. Basic CSS Optimization (Autoprefixer + Minify)**
function optimizeCSS() {
  return gulp.src(paths.css.src)
    .pipe(sourcemaps.init())
    .pipe(postcss([
      autoprefixer(),  // Auto-add vendor prefixes
      cssnano()       // Minify CSS
    ]))
    .pipe(rename({ suffix: '.min' }))
    .pipe(sourcemaps.write('.'))
    .pipe(gulp.dest(paths.css.dest));
}

// **2. Concatenate + Optimize All CSS into One File**
function concatCSS() {
  return gulp.src(paths.css.src)
    .pipe(sourcemaps.init())
    .pipe(concat('styles.min.css'))  // Merges all CSS into one file
    .pipe(postcss([
      autoprefixer(),
      cssnano()
```

```
    ]))
    .pipe(sourcemaps.write('.'))
    .pipe(gulp.dest(paths.css.dest));
}

// **3. Advanced CSS Optimization (Sorting, Media Queries, Duplicates)**
function advancedCSS() {
  const plugins = [
    autoprefixer(),
    sorting({ 'properties-order': 'alphabetical' }), // Sort CSS properties
    combineMediaQueries(),  // Combine duplicate media queries
    combineSelectors({ removeDuplicatedProperties: true }), // Remove duplicate
selectors
    cssnano({ preset: 'advanced' })  // Extreme minification
  ];

  return gulp.src(paths.css.src)
    .pipe(sourcemaps.init())
    .pipe(postcss(plugins))
    .pipe(rename({ suffix: '.advanced' }))
    .pipe(sourcemaps.write('.'))
    .pipe(gulp.dest(paths.css.dest));
}

// **4. Watch for Changes (Development)**
function watchCSS() {
  gulp.watch(paths.css.src, optimizeCSS);
}

// **Export Tasks**
exports.default = optimizeCSS;  // Run with `gulp`
exports.concat = concatCSS;     // Run with `gulp concat`
exports.advanced = advancedCSS; // Run with `gulp advanced`
exports.watch = watchCSS;      // Run with `gulp watch`
```

Q 2. Show the use of service worker on same website

Index.php

```
<script>
    if ('serviceWorker' in navigator) {
      navigator.serviceWorker
        .register('./ws.js') // ☑ Correct path
        .then(reg => console.log('☑ SW Registered! Scope:', reg.scope))
        .catch(err => console.error('❌ SW Failed:', err));
    }
  </script>
```

Script.js

```
document.addEventListener('DOMContentLoaded', () => {
  if ('serviceWorker' in navigator) {
    const registerSW = async () => {
      try {
        const registration = await navigator.serviceWorker.register('./ws.js', {
          scope: './',
          updateViaCache: 'none'
        });
```

```
      console.log('✅ SW registered. Scope:', registration.scope);
      registration.update();

      // Track SW updates
      registration.addEventListener('updatefound', () => {
        const newWorker = registration.installing;
        newWorker.addEventListener('statechange', () => {
          console.log(`SW state: ${newWorker.state}`);
        });
      });

    } catch (error) {
      console.error('❌ SW registration failed:', error);
      if (error.message.includes('404')) {
        alert('Service worker file (ws.js) not found!');
      }
    }
  };

  setTimeout(registerSW, 500); // Prevent race conditions
}
});
```

Ws.js

```
const CACHE_NAME = 'hopevein-cache-final-v1';

const OFFLINE_PAGE = './index.html';

const urlsToCache = [

  './',

  './index.html',

  './css/style.css',

  './css/about.css',

  './css/contact.css',

  './script.js',

  './about.html',

  './contact.html',

  './courses.html',

  './images/avatar1.jpg',

  './images/avatar2.jpg',

  './images/avatar3.jpg',

  './images/avatar4.jpg',

  './images/avatar5.jpg',

  './images/avatar6.jpg',

  './images/avatar7.jpg',

  './images/course1.jpg',
```

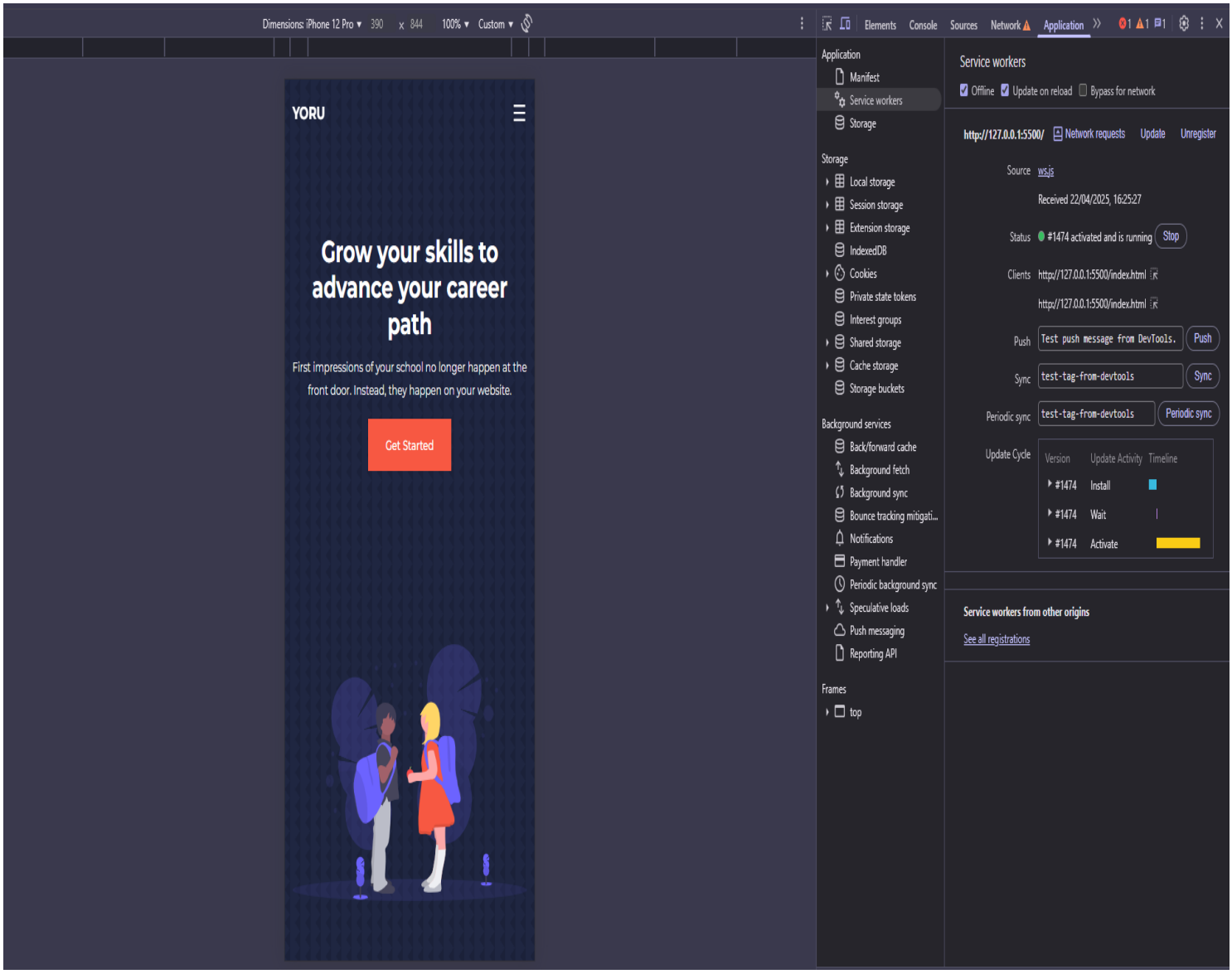
```
  './images/course2.jpg',
  './images/course3.jpg',
  './images/course4.jpg',
  './images/course5.jpg',
  './images/course6.jpg',
  './images/course7.jpg',
  './images/course8.jpg',
  './images/course9.jpg',
  './images/course10.jpg',
  './images/course12.jpg',
  './images/course13.jpg',
  './images/course14.jpg',
  './images/course15.jpg',
  './images/course16.jpg',
  './images/course17.jpg',
];

// Install event - cache all critical resources
self.addEventListener('install', event => {
  event.waitUntil(
    caches.open(CACHE_NAME)
      .then(cache => {
        console.log('[SW] Cache opened:', CACHE_NAME);
        return cache.addAll(urlsToCache.map(url => new Request(url, {
          cache: 'reload',
          credentials: 'same-origin'
        })));  // ✅ Fixed missing parenthesis here
```

```
      })
      .then(() => {
        console.log('[SW] All resources cached');
        return self.skipWaiting();
      })
  );
});

// Rest of the code remains unchanged...
```

OUTPUT:-



Work on offline mode