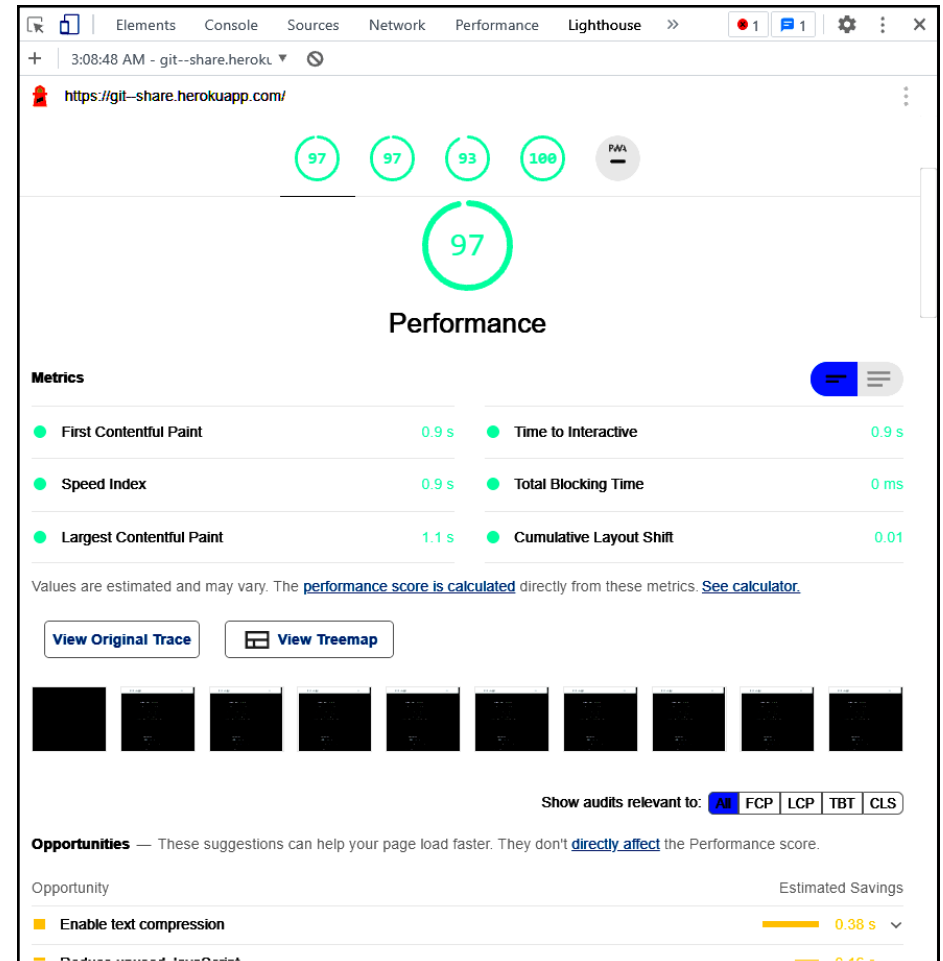
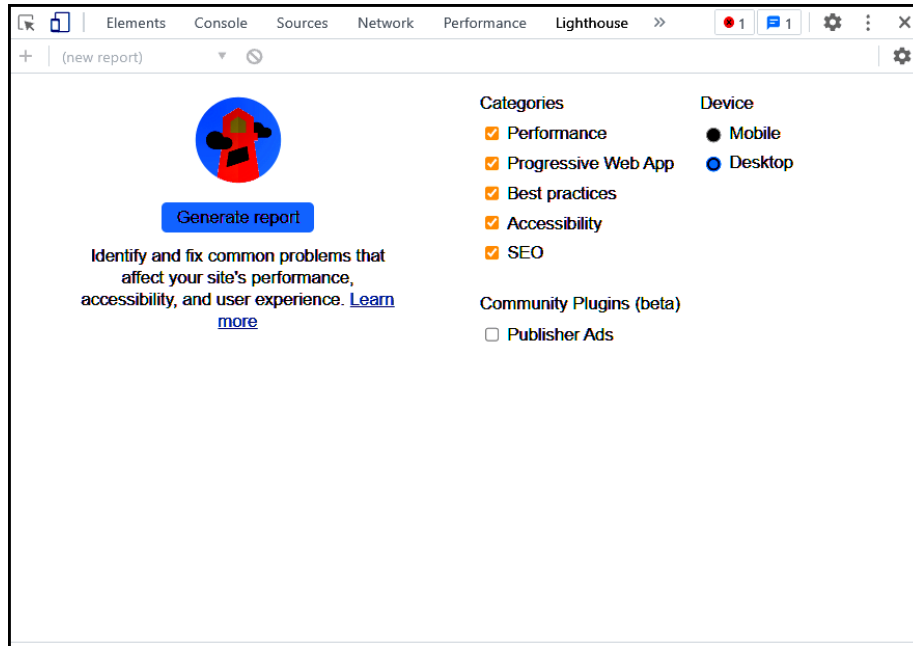


Lighthouse



Minify

← → ↺ 🏠 jscompress.com ☆

Copy & Paste JavaScript Code Upload Javascript Files Output ☐ ECMAScript 2021 (via babel-minify)

```
const path = require("path");
const express = require("express");
const session = require("express-session");
const exphbs = require("express-handlebars");
const routes = require("./controllers");
const helpers = require("./utils/helpers");
var compression = require("compression");

const sequelize = require("./config/connection");
const SequelizeStore = require("connect-session-sequelize")(session.Store);

const app = express();
const PORT = process.env.PORT || 3001;

// Set up Handlebars.js engine with custom helpers
const hbs = exphbs.create({ helpers });

const sess = {
  secret: "Super secret secret".
```

Compress JavaScript

About JSCompress
JSCompress is an online JavaScript compressor that allows you to compress and minify all of your JS files by up to 80% of their original size. Copy and paste your code or you can upload and combine multiple files and then compress. We use [UglifyJS 3](#) and [babel-minify](#) for all JavaScript minification and compression.

← → ↺ 🏠 jscompress.com ☆

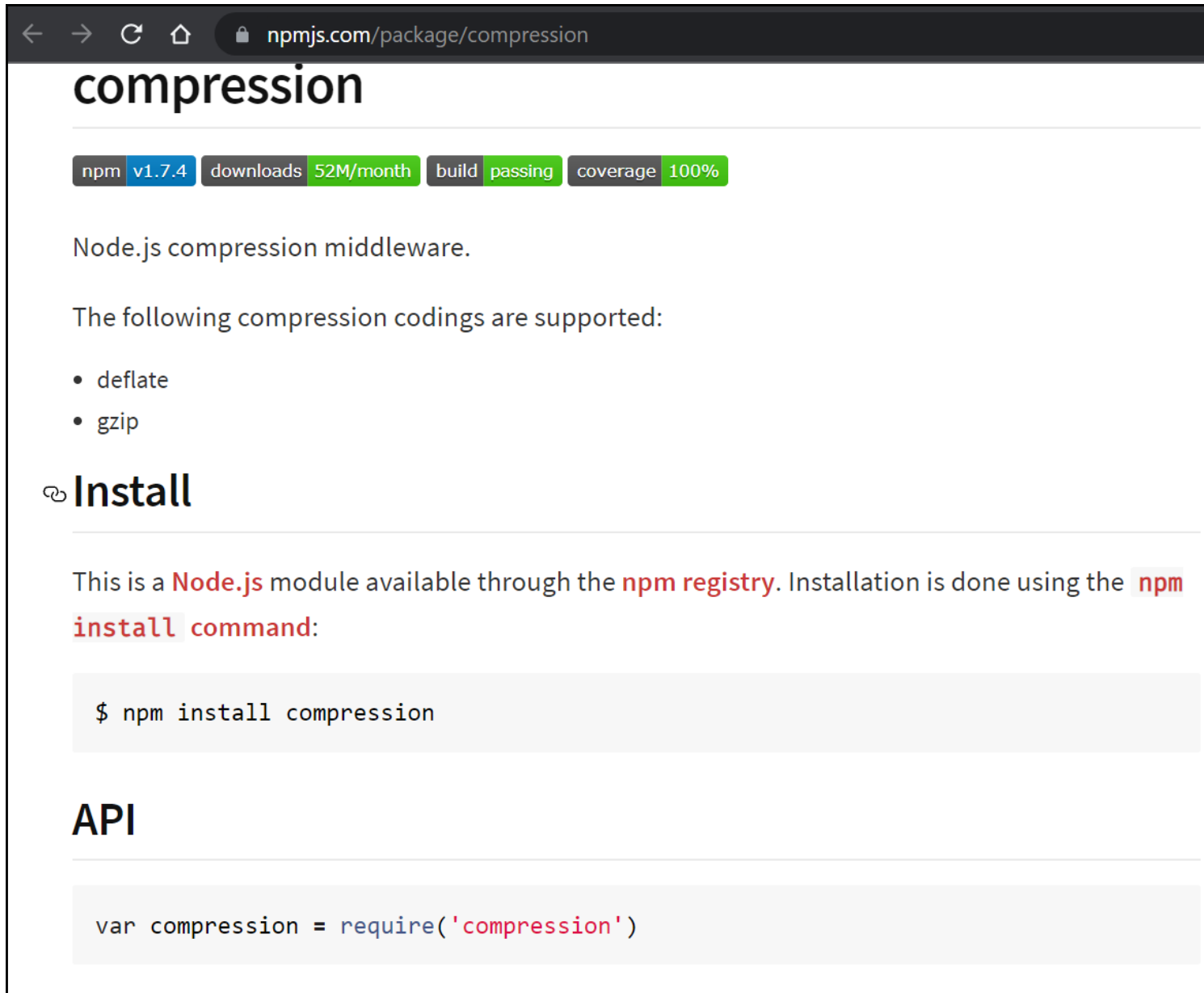
Copy & Paste JavaScript Code Upload Javascript Files Output ☒ ECMAScript 2021 (via babel-minify)

```
const path=require("path"),express=require("express"),session=require("express-session"),exphbs=require("express-handlebars"),routes=require("./controllers"),helpers=require("./utils/helpers");var compression=require("compression");const sequelize=require("./config/connection"),SequelizeStore=require("connect-session-sequelize")(session.Store),app=express(),PORT=process.env.PORT||3001,hbs=exphbs.create({helpers}),sess={secret:"Super secret secret",cookie:{},resave:!1,saveUninitialized:!0,store:new SequelizeStore({db:sequelize})};app.use(session(sess)),app.engine("handlebars",hbs.engine),app.set("view engine","handlebars"),app.use(express.json()),app.use(express.urlencoded({extended:!0})),app.use(express.static(path.join(__dirname,"public"))),app.use(routes),sequelize.sync({force:!1}).then(()=>{app.listen(PORT,()=>console.log(`\n----|----- Now listening at port ${PORT} ----|-----`)}));
```

Download Stats: 24.58 % compression, saving 0.29 kb

4	express = requ
5	routes = requ
6	helpers = req
7	sequelize = r
8	SequelizeStor
9	app = express
10	PORT = proces
11	hbs = exphbs.
12	sess = {
13	secret: "Su

Compression



The screenshot shows the npmjs.com page for the 'compression' package. The browser address bar displays 'npmjs.com/package/compression'. The package name 'compression' is prominently displayed at the top. Below it, a row of badges provides key statistics: 'npm v1.7.4', 'downloads 52M/month', 'build passing', and 'coverage 100%'. The description states 'Node.js compression middleware.' and lists supported compression codings as 'deflate' and 'gzip'. An 'Install' section, indicated by a link icon, explains that this is a Node.js module available through the npm registry and provides the command '\$ npm install compression'. An 'API' section follows, showing the code 'var compression = require('compression')'.

← → ↻ 🏠 npmjs.com/package/compression

compression

npm v1.7.4 downloads 52M/month build passing coverage 100%

Node.js compression middleware.

The following compression codings are supported:

- deflate
- gzip

🔗 Install






This is a **Node.js** module available through the **npm registry**. Installation is done using the **npm install** command:

```
$ npm install compression
```


API

```
var compression = require('compression')
```


Image Compression

    [tinypng.com](#) 

WEB **BETA** CDN PHOTOSHOP DEVELOPER API THIRD-PARTY ANALYZER [LOGIN](#)






Drop your WebP, PNG or JPEG files here!
Up to 20 images, max 5 MB each.



1.jpg	206.2 KB	Finished	108.8 KB	download	-47%
2.jpg	186.3 KB	Finished	100.8 KB	download	-46%
3.jpg	108.7 KB	Finished	69.1 KB	download	-36%
4.jpg	101.8 KB	Finished	62.2 KB	download	-39%
5.jpg	19.4 KB	Finished	13.1 KB	download	-33%
6.jpg	322.2 KB	Finished	156.3 KB	download	-51%
7.jpg	113.9 KB	Finished	72.2 KB	download	-37%
8.jpg	100.7 KB	Compressing			

32.jpg	376.9 KB	Finished	207.7 KB	download	-45%
33.jpg	48.1 KB	Finished	28.5 KB	download	-41%
37.jpg	200.1 KB	Finished	98.5 KB	download	-51%
38.jpg	147.4 KB	Finished	72.1 KB	download	-51%
31.jpg	271.9 KB	Finished	140.0 KB	download	-49%



 Save to Dropbox  Download all

Panda just saved you **48%** 498 KB total

Lazy Load

```
loadImages();
```

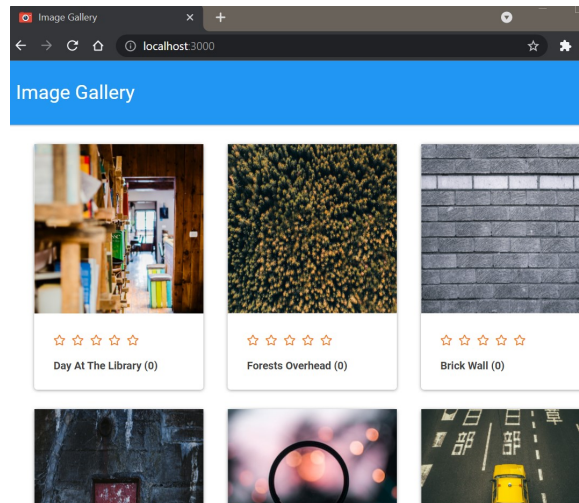
```
function createEl(htmlString = "", className) {
  const el = document.createElement(htmlString);
  if (className) {
    el.setAttribute("class", className);
  }
  return el;
}
```

```
function initLazyImages() {
  const lazyImages = document.querySelectorAll(".lazy-image");
  function onIntersection(imageEntities) {
    imageEntities.forEach(image => {
      if (image.isIntersecting) {
        observer.unobserve(image.target);
        image.target.src = image.target.dataset.src;
      }
    });
  }
  const observer = new IntersectionObserver(onIntersection);
  lazyImages.forEach(image => observer.observe(image));
}
```

```
function loadImages() {
  fetch("/api/images")
    .then(res => res.json())
    .then(data => createCards(data))
    .then(() => initLazyImages());
}
```

```
function createCards(data) {
  const container = document.querySelector(".container");
  container.innerHTML = "";
  let lastRow;
  const row = createEl("div", "row");
  return data.forEach(function(image, index) {
    const col = createEl("div", "col-md-4 mt-4");
    col.appendChild(createCard(image));
    if (index % 3 === 0) {
      row.appendChild(col);
      container.appendChild(row);
      lastRow = row;
    }
    return lastRow.appendChild(col);
  });
}
```

```
function createCard(image) {
  const card = createEl("div", "card");
  const imageContainer = createEl("div",
    "card__image-container");
  const img = createEl("img",
    "card-img-top card__image--cover lazy-image");
  img.setAttribute(
    "src",
    "data:image/png;base64,iVBORw0KGgoAAAANSUUhEUg
    AAAAEAAAABCAQAAAC1HawCAAAAC
    0lEQVR42mOMrgcAATsA3BT31OAAAAA
    SUVORK5CYII=");
  img.setAttribute("data-src", image.image);
  img.setAttribute("alt", image.description);
  const cardBody = createEl("div", "card-body");
  const ratingFormContainer = createEl("div",
    "rating d-flex justify-content-start");
  ratingFormContainer.setAttribute("data-id", image._id);
  ratingFormContainer.setAttribute("data-rating", image.rating);
  const ratingForm = createRatingForm(image);
  const cardText = createEl("p",
    "card-text font-weight-bold mt-2");
  cardText.innerHTML = `${image.description} (${image.rating})`;
  imageContainer.appendChild(img);
  ratingFormContainer.appendChild(ratingForm);
  cardBody.appendChild(ratingFormContainer);
  cardBody.appendChild(cardText);
  card.appendChild(imageContainer);
  card.appendChild(cardBody);
  return card;
}
```



```
function createRatingForm(image) {
  const labelText = {
    1: "One Star",
    2: "Two Stars",
    3: "Three Stars",
    4: "Four Stars",
    5: "Five Stars"
  };
  const form = createEl("form");
  form.setAttribute("action", "post");
  for (let i = 1; i <= 5; i++) {
    const input = createEl("input", "visuallyhidden");
    input.setAttribute("type", "radio");
    input.setAttribute("name", "rating");
    input.setAttribute("id", `${image._id}-star-${i}`);
    input.setAttribute("value", i);
    const label = createEl("label");
    label.setAttribute("for", `${image._id}-star-${i}`);
    const labelSpan = createEl("span", "visuallyhidden");
    labelSpan.innerText = labelText[i];
    const star = createEl("i", `fa-star ${image.rating >= i ?
      "fas" : "far"}`);
    label.appendChild(labelSpan);
    label.appendChild(star);
    label.onclick = updateRating;
    form.appendChild(input);
    form.appendChild(label);
  }
  return form;
}

function updateRating(event) {
  const [id, , rating] = event.currentTarget.getAttribute("for")
    .split("-");
  fetch(`/api/images/${id}`, {
    method: "PUT",
    body: JSON.stringify({ rating }),
    headers: {
      "Content-Type": "application/json"
    }
  }).then(function() {
    loadImages();
  });
}
```

```
var express = require("express");
var mongoose = require("mongoose");
```

```
// import the compression package
var compression = require("compression");
```

```
var app = express();
var PORT = process.env.PORT || 3000;
```

```
// enable compression middleware
app.use(compression());
```

```
app.use(express.static("public"));
app.use(express.urlencoded({ extended: true }));
app.use(express.json());
require("./routes/apiRoutes")(app);
require("./routes/htmlRoutes")(app);
mongoose.connect(process.env.MONGODB_URI || "mongodb://localhost/
imageperformance", {
  useNewUrlParser: true
});
app.listen(PORT, function() {
  console.log(`Now listening on port: ${PORT}`);
});
```

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <meta http-equiv="X-UA-Compatible" content="ie=edge" />
    <link
      rel="stylesheet"
      href="https://cdn.jsdelivr.net/npm/bootstrap/4.3.1/
        materia/bootstrap.min.css"
    />
    <link
      rel="stylesheet"
      href="https://use.fontawesome.com/releases/v5.8.2/css/all.css"
      integrity="sha384-oS3vJWv+0UjzBfQzYUhtDYW+
        Pj2yciDJxpsK1OYPAYjqT085Qq/1cq5FLXAZQ7Ay"
      crossorigin="anonymous"
    />
    <link rel="stylesheet" href="/assets/css/style.css" />
    <title>Image Gallery</title>
  </head>
  <body>
    <nav class="navbar navbar-dark bg-primary">
      <a class="navbar-brand" href="#">Image Gallery</a>
    </nav>
    <div class="container"></div>
    <!-- link to minified javascript file -->
    <script src="/dist/index.js"></script>
  </body>
</html>
```

Gallery Optimize

```
loadImages();

function createEl(htmlString = "", className) {
  const el = document.createElement(htmlString);
  if (className) {
    el.setAttribute("class", className);
  }
  return el;
}

function initLazyImages() {
  const lazyImages = document.querySelectorAll(".lazy-image");
  function onIntersection(imageEntities) {
    imageEntities.forEach(image => {
      if (image.isIntersecting) {
        observer.unobserve(image.target);
        image.target.src = image.target.dataset.src;
      }
    });
  }
  const observer = new IntersectionObserver(onIntersection);
  lazyImages.forEach(image => observer.observe(image));
}

function loadImages() {
  fetch("/api/images")
    .then(res => res.json())
    .then(data => createCards(data))
    .then(() => initLazyImages());
}

function createCards(data) {
  const container = document.getElementsByClassName(
    "container"
  )[0];
  container.innerHTML = "";
  let lastRow;
  const row = createEl("div", "row");
  return data.forEach(function(image, index) {
    const col = createEl("div", "col-md-4 mt-4");
    col.appendChild(createCard(image));
    if (index % 3 === 0) {
      row.appendChild(col);
      container.appendChild(row);
      lastRow = row;
    }
    return lastRow.appendChild(col);
  });
}

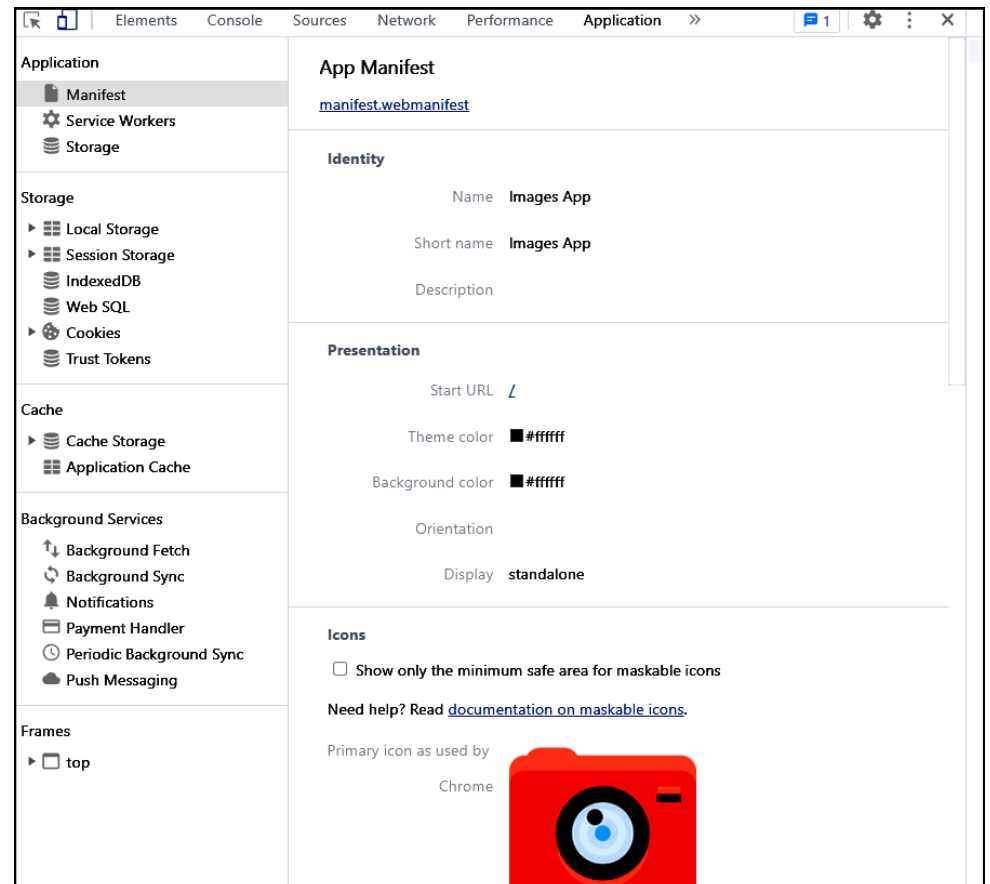
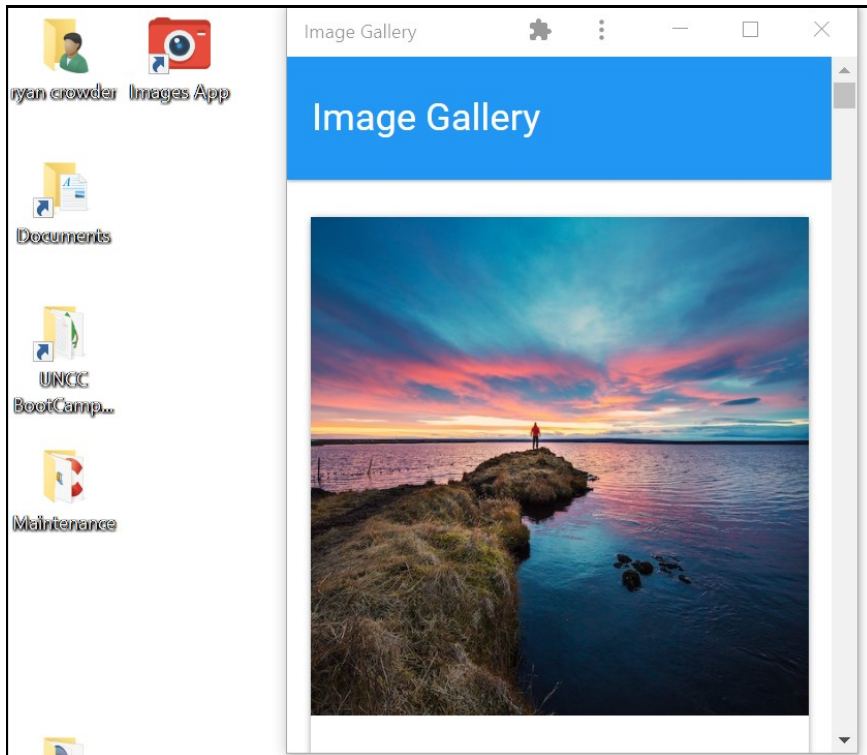
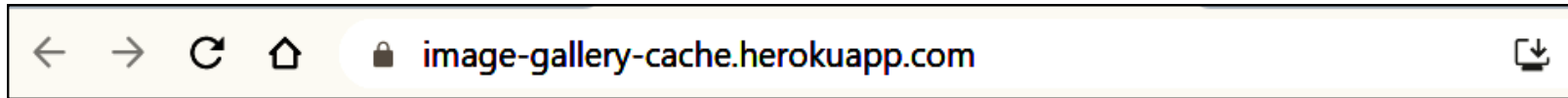
function createCard(image) {
  const card = createEl("div", "card");
  const imageContainer = createEl("div", "card__image-container");
  const img = createEl("img",
    "card-img-top card__image--cover lazy-image");
  img.setAttribute(
    "src",
    "data:image/png;base64,iVBORw0KGgoAAAANSUuEUg
      AAAAEAAAAACAQAAAC1HAWCAAAAC0IE
      QVR42mOMrgcAATsA3BT3IOAAAAASU
      VORK5CYII="
  );
  img.setAttribute("data-src", image.image);
  img.setAttribute("alt", image.description);
  const cardBody = createEl("div", "card-body");
```

```
const ratingFormContainer = createEl("div",
  "rating d-flex justify-content-start");
ratingFormContainer.setAttribute("data-id", image._id);
ratingFormContainer.setAttribute("data-rating", image.rating);
const ratingForm = createRatingForm(image);
const cardText = createEl("p", "card-text font-weight-bold mt-2");
cardText.innerText = `${image.description} (${image.rating})`;
imageContainer.appendChild(img);
ratingFormContainer.appendChild(ratingForm);
cardBody.appendChild(ratingFormContainer);
cardBody.appendChild(cardText);
card.appendChild(imageContainer);
card.appendChild(cardBody);
return card;
}

function createRatingForm(image) {
  const labelText = {
    1: "One Star",
    2: "Two Stars",
    3: "Three Stars",
    4: "Four Stars",
    5: "Five Stars"
  };
  const form = createEl("form");
  form.setAttribute("action", "post");
  for (let i = 1; i <= 5; i++) {
    const input = createEl("input", "visuallyhidden");
    input.setAttribute("type", "radio");
    input.setAttribute("name", "rating");
    input.setAttribute("id", `${image._id}-star-${i}`);
    input.setAttribute("value", i);
    const label = createEl("label");
    label.setAttribute("for", `${image._id}-star-${i}`);
    const labelSpan = createEl("span", "visuallyhidden");
    labelSpan.innerText = labelText[i];
    const star = createEl("i", `fa-star ${image.rating >= i ?
      "fas" : "far"}`);
    label.appendChild(labelSpan);
    label.appendChild(star);
    label.onclick = updateRating;
    form.appendChild(input);
    form.appendChild(label);
  }
  return form;
}

function updateRating(event) {
  const [id, , rating] = event.currentTarget.getAttribute("for")
    .split("-");
  fetch(`/api/images/${id}`, {
    method: "PUT",
    body: JSON.stringify({ rating }),
    headers: {
      "Content-Type": "application/json"
    }
  }).then(function() {
    loadImages();
  });
}
```

PWA



Manifest

```
{
  "short_name": "Demo",
  "name": "Web App Manifest Demo",
  "icons": [
    {
      "src": "/assets/images/icons-192.png",
      "sizes": "192x192",
      "type": "image/png"
    },
    {
      "src": "/assets/images/icons-512.png",
      "sizes": "512x512",
      "type": "image/png"
    }
  ],
  "start_url": "/",
  "background_color": "#808080",
  "display": "standalone",
  "theme_color": "#808080"
}
```

```
▼ models
  JS image.js
  JS index.js
▼ public
  > assets
  ★ favicon.ico
  <> index.html
  {} manifest.webmanifest
  > routes
  > seeders
  .gitignore
  {} package.json
  JS server.js
```

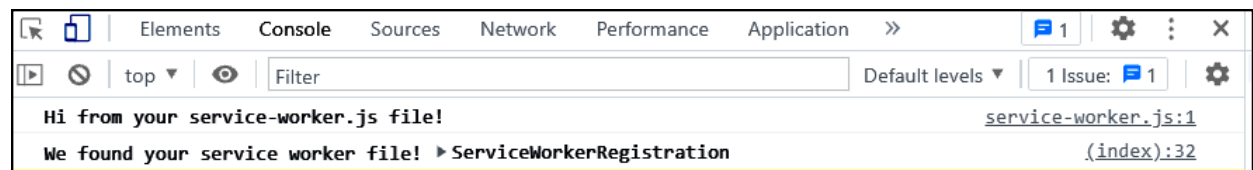
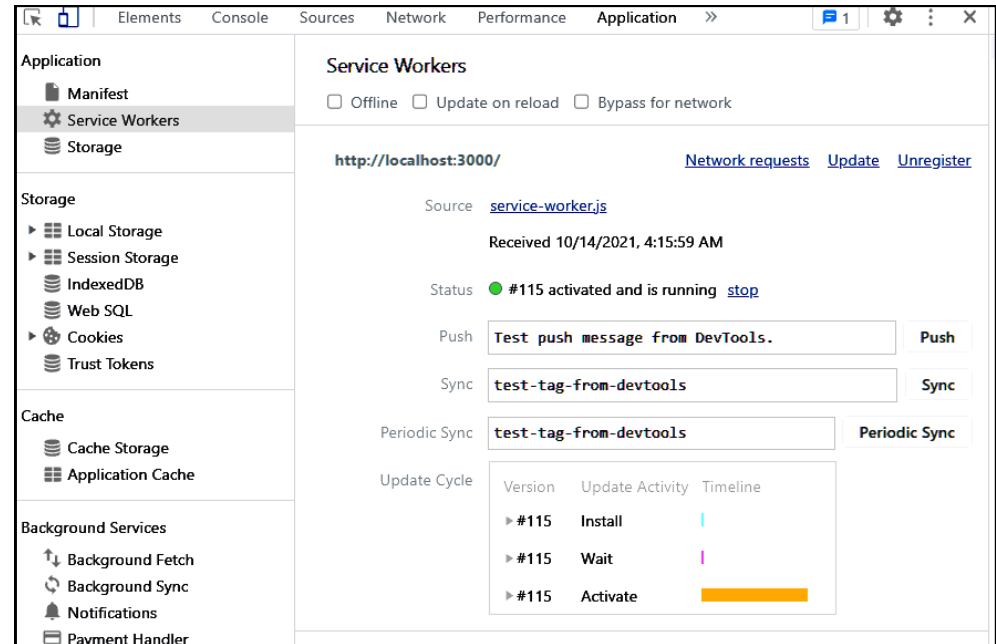
```
{
  "name": "Images App",
  "short_name": "Images App",
  "icons": [
    {
      "src": "assets/images/icons/icon-72x72.png",
      "sizes": "72x72",
      "type": "image/png"
    },
    {
      "src": "assets/images/icons/icon-96x96.png",
      "sizes": "96x96",
      "type": "image/png"
    },
    {
      "src": "assets/images/icons/icon-128x128.png",
      "sizes": "128x128",
      "type": "image/png"
    },
    {
      "src": "assets/images/icons/icon-144x144.png",
      "sizes": "144x144",
      "type": "image/png"
    },
    {
      "src": "assets/images/icons/icon-152x152.png",
      "sizes": "152x152",
      "type": "image/png"
    },
    {
      "src": "assets/images/icons/icon-192x192.png",
      "sizes": "192x192",
      "type": "image/png"
    },
    {
      "src": "assets/images/icons/icon-384x384.png",
      "sizes": "384x384",
      "type": "image/png"
    },
    {
      "src": "assets/images/icons/icon-512x512.png",
      "sizes": "512x512",
      "type": "image/png"
    }
  ],
  "theme_color": "#ffffff",
  "background_color": "#ffffff",
  "start_url": "/",
  "display": "standalone"
}
```


Service Worker

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <meta http-equiv="X-UA-Compatible" content="ie=edge" />
  <link
    rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/bootswatch/4.3.1/
      materia/bootstrap.min.css"
  />
  <link
    rel="stylesheet"
    href="https://use.fontawesome.com/releases/v5.8.2/css/all.css"
    integrity="sha384-oS3vJWv+0UjzBfQzYUhtDYW+
      Pj2yciDJxpsK1OYPAYjqT085Qq/1cq5FLXAZQ7Ay"
    crossorigin="anonymous"
  />
  <link rel="stylesheet" href="/assets/css/style.css" />
  <link rel="manifest" href="manifest.webmanifest" />
  <title>Image Gallery</title>
</head>
<body>
  <nav class="navbar navbar-dark bg-primary">
    <a class="navbar-brand" href="#">Image Gallery</a>
  </nav>
  <div class="container"></div>
  <script src="/assets/js/loadImages.js"></script>
```

```
<script>
if ("serviceWorker" in navigator) {
  window.addEventListener("load", () => {
    navigator.serviceWorker.register("service-worker.js").then((reg) => {
      console.log("We found your service worker file!", reg);
    });
  });
}
</script>
</body>
</html>
```

```
console.log("Hi from your service-worker.js file!");
```



Caching/Fetching Files

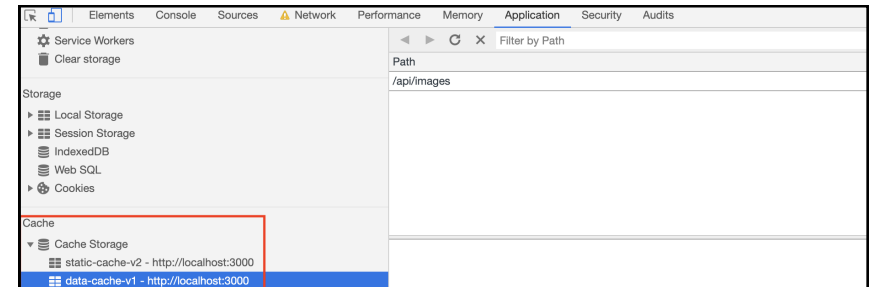
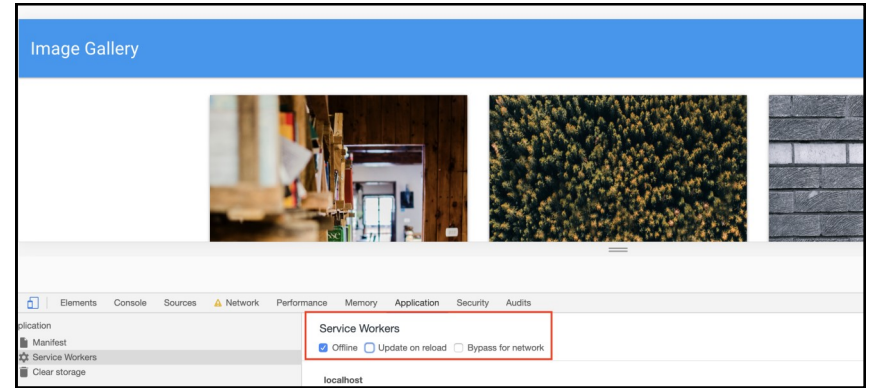
```
const CACHE_NAME = "static-cache-v2";
const DATA_CACHE_NAME = "data-cache-v1";
```

```
const FILES_TO_CACHE = [
  "/",
  "/index.html",
  "/favicon.ico",
  "/manifest.webmanifest",
  "/assets/css/style.css",
  "/assets/js/loadImages.js",
  "/assets/images/icons/icon-72x72.png",
  "/assets/images/icons/icon-96x96.png",
  "/assets/images/icons/icon-128x128.png",
  "/assets/images/icons/icon-144x144.png",
  "/assets/images/icons/icon-152x152.png",
  "/assets/images/icons/icon-192x192.png",
  "/assets/images/icons/icon-384x384.png",
  "/assets/images/icons/icon-512x512.png",
  "/assets/images/1.jpg",
  "/assets/images/2.jpg",
  "/assets/images/3.jpg",
  "/assets/images/4.jpg",
  "/assets/images/5.jpg",
  "/assets/images/6.jpg",
  "/assets/images/7.jpg",
  "/assets/images/8.jpg",
  "/assets/images/9.jpg",
  "/assets/images/10.jpg",
  "/assets/images/11.jpg",
  "/assets/images/12.jpg",
  "/assets/images/13.jpg",
  "/assets/images/14.jpg",
  "/assets/images/15.jpg",
  "/assets/images/16.jpg",
  "/assets/images/17.jpg",
  "/assets/images/18.jpg",
  "/assets/images/19.jpg",
  "/assets/images/20.jpg",
  "/assets/images/21.jpg",
  "/assets/images/22.jpg",
  "/assets/images/23.jpg",
  "/assets/images/24.jpg",
  "/assets/images/25.jpg",
  "/assets/images/26.jpg",
  "/assets/images/27.jpg",
  "/assets/images/28.jpg",
  "/assets/images/29.jpg",
  "/assets/images/30.jpg",
  "/assets/images/31.jpg",
  "/assets/images/32.jpg",
  "/assets/images/33.jpg",
  "/assets/images/34.jpg",
  "/assets/images/35.jpg",
  "/assets/images/36.jpg",
  "/assets/images/37.jpg",
  "/assets/images/38.jpg",
];
```

```
// install
self.addEventListener("install", function (evt) {
  // pre cache image data
  evt.waitUntil(
    caches.open(DATA_CACHE_NAME).then((cache) =>
      cache.add("/api/images")
    );
  // pre cache all static assets
  evt.waitUntil(
    caches.open(CACHE_NAME).then((cache) =>
      cache.addAll(FILES_TO_CACHE)
    );
  // tell the browser to activate this service worker immediately
  // once it has finished installing
  self.skipWaiting();
});
```

```
self.addEventListener("activate", function (evt) {
  evt.waitUntil(
    caches.keys().then((keyList) => {
      return Promise.all(
        keyList.map((key) => {
          if (key !== CACHE_NAME && key !==
            DATA_CACHE_NAME) {
            console.log("Removing old cache data", key);
            return caches.delete(key);
          }
        })
      );
    });
  self.clients.claim();
});
```

```
self.addEventListener("fetch", function (evt) {
  // code to handle requests goes here
  if (evt.request.url.includes("api")) {
    console.log("[Service Worker] Fetch (data", evt.request.url);
    evt.respondWith(
      caches.open(DATA_CACHE_NAME).then((cache) => {
        return fetch(evt.request)
          .then((response) => {
            if (response.status === 200) {
              cache.put(evt.request.url, response.clone());
            }
            return response;
          })
          .catch((err) => {
            return cache.match(evt.request);
          });
      });
    );
  }
  return;
});
evt.respondWith(
  caches.open(CACHE_NAME).then((cache) => {
    return cache.match(evt.request).then((response) => {
      return response || fetch(evt.request);
    });
  });
);
```



Path	Response-Type	Content-Type
/		basic, text/html; charset=UTF-8
/assets/css/style.css		basic, text/css; charset=UTF-8
/assets/images/1.jpg		basic, image/jpeg
/assets/images/10.jpg		basic, image/jpeg
/assets/images/11.jpg		basic, image/jpeg
/assets/images/12.jpg		basic, image/jpeg
/assets/images/13.jpg		basic, image/jpeg
/assets/images/14.jpg		basic, image/jpeg
/assets/images/15.jpg		basic, image/jpeg

```
const CACHE_NAME = "static-cache-v2";
const DATA_CACHE_NAME = "data-cache-v1";
```

```
const FILES_TO_CACHE = [
  "/",
  "/index.html",
  "/manifest.webmanifest",
  "/assets/css/style.css",
  "/assets/js/loadImages.js",
  "/assets/images/brandenburg.jpg",
  "/assets/images/logo16.jpg",
  "/assets/images/logo48.jpg",
  "/assets/images/logo128.jpg",
  "/assets/images/logo512.jpg",
  "/assets/images/map.jpg",
  "/assets/images/reichstag.jpg",
];

// install
self.addEventListener("install", function (evt) {
  // pre cache image data
  evt.waitUntil(
    caches
      .open(DATA_CACHE_NAME)
      .then((cache) => cache.add("./unsolved/assets/images"))
  );

  // pre cache all static assets
  evt.waitUntil(
    caches.open(CACHE_NAME).then((cache) => cache.addAll(FILES_TO_CACHE))
  );
  // tell the browser to activate this service worker immediately once it
  // has finished installing
  self.skipWaiting();
});
```

```
self.addEventListener("activate", function (evt) {
  evt.waitUntil(
    caches.keys().then((keyList) => {
      return Promise.all(
        keyList.map((key) => {
          if (key !== CACHE_NAME && key !== DATA_CACHE_NAME) {
            console.log("Removing old cache data", key);
            return caches.delete(key);
          }
        })
      );
    })
  );
  self.clients.claim();
});
```

```
self.addEventListener("fetch", function (evt) {
  // code to handle requests goes here
  if (evt.request.url.includes("api")) {
    console.log("[Service Worker] Fetch (data", evt.request.url);
    evt.respondWith(
      caches.open(DATA_CACHE_NAME).then((cache) => {
        return fetch(evt.request)
          .then((response) => {
            if (response.status === 200) {
              cache.put(evt.request.url, response.clone());
            }
            return response;
          })
          .catch((err) => {
            return cache.match(evt.request);
          })
      );
    );
    return;
  }
  evt.respondWith(
    caches.open(CACHE_NAME).then((cache) => {
      return cache.match(evt.request).then((response) => {
        return response || fetch(evt.request);
      });
    })
  );
});
```

Map PWA

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <meta name="theme-color" content="#009578" />
  <title>Map PWA</title>
  <!-- TODO: Add link for manifest -->
  <link rel="manifest" href="manifest.webmanifest" />
  <link rel="stylesheet" href="/assets/css/style.css" />
  <link rel="apple-touch-icon" href="/assets/images/logo128.png" />
</head>
<body>
  <section class="container">
    <h2>Trip Planner</h2>
    <div class="agenda">
      <h4>Getting Situated</h4>
      <table class="sites">
        <tr>
          <td>
            <a href="https://24timezones.com/Berlin/time#gref" target="_blank">
              Local Time</a>
          </td>
          <td>
            <a href="https://www.international.hu-berlin.de/en/studierende/aus-dem-ausland/wegweiser/11/index" target="_blank">
              Emergency Services</a>
          </td>
        </tr>
        <tr>
          <td>
            <a href="https://en.wikipedia.org/wiki/Airports_of_Berlin" target="_blank">
              Airport</a>
          </td>
        </tr>
      </table>
    </div>
    <div>
      
    </section>
    <section class="container">
      <div class="agenda">
        <h4>Places to See</h4>
        <table class="sites">
          <tr>
            <td>
              <a href="https://en.wikipedia.org/wiki/Brandenburg" target="_blank">
                Brandenburg Gate</a>
            </td>
          </tr>
        </table>
      </div>
      
    </section>
    <section class="container">
      <div class="agenda">
        <h4>Places to See</h4>
        <table class="sites">
          <tr>
            <td>
              <a href="https://en.wikipedia.org/wiki/Reichstag_building" target="_blank">
                Reichstag</a>
            </td>
          </tr>
        </table>
      </div>
      
    </section>
    <script src="/service-worker.js"></script>
  </script>
  if ("serviceWorker" in navigator) {
    navigator.serviceWorker
      .register("service-worker.js")
      .then((registration) => {
        console.log("Service Worker Registered!");
      })
      .catch((error) => {
        console.log("Service Worker Registration Failed!");
        console.log(error);
      });
  }
</script>
</body>
</html>
```

```
{
  "name": "Map PWA",
  "short_name": "Map PWA",
  "start_url": ".",
  "background_color": "#cadbda",
  "theme_color": "#cadbda",
  "display": "standalone",
  "icons": [
    {
      "src": "assets/images/logo16.png",
      "sizes": "16x16",
      "type": "image/png"
    },
    {
      "src": "assets/images/logo48.png",
      "sizes": "48x48",
      "type": "image/png"
    },
    {
      "src": "assets/images/logo128.png",
      "sizes": "128x128",
      "type": "image/png"
    },
    {
      "src": "assets/images/logo512.png",
      "sizes": "512x512",
      "type": "image/png"
    }
  ]
}
```

Webpack

Webpack

```
* ATTENTION: The "eval" devtool has been used (maybe by default in mode: "development").
* This devtool is neither made for production nor for readable output files.
* It uses "eval()" calls to create a separate source file in the browser devtools.
* If you are trying to read the output file, select a different devtool (https://webpack.js.org/configuration/devtool/)
* or disable the default devtool with "devtool: false".
* If you are looking for production-ready output files, see mode: "production" (https://webpack.js.org/configuration/mode/).
*/
/******/ (O=> { // webpackBootstrap
/******/ var __webpack_modules__ = ( {
/***/ "./src/app.js":
/*!*****!*\
*** ./src/app.js ***!
*****/
/***/ ((__unused_webpack_module, __unused_webpack_exports, __webpack_require__) => {
eval("const priceEl = document.getElementById('price');\nconst balanceEl = document.getElementById('balance');\n\nconst expenseEl = document.getElementById('expense');\nconst expensesListEl = document.getElementById('expenses-list');\n\nconst submitBtn = document.getElementById('submit');\nconst resetBtn = document.getElementById('reset');\n\nconst subtract = __webpack_require__('./src sync recursive')(calculation.js);\n\nfunction addToList(name, price) {\n  expensesListEl.innerHTML += `<li class='list-group-item'>Name: $ {name}\n    <span class='ml-4'>Price: $ {price}</span></li>`; }\n\nfunction submit(e) {\n  e.preventDefault();\n  addToList(expenseEl.value, priceEl.value); }\n\nfunction reset(e) {\n  e.preventDefault(); }\n\nsubmitBtn.onclick = submit;\nresetBtn.onclick = reset;\n\n\n// sourceURL=webpack://webpack-demo./src/app.js?");
/***/ }),
/***/ "./src sync recursive":
/*!*****!*\
*** ./src/ sync ***!
*****/
/***/ ((module) => {
eval("function webpackEmptyContext(req) {\ntvar e = new Error(`Cannot find module ` + req + ``);\nte.code = 'MODULE_NOT_FOUND';\ntthrow e;\n}\n\nwebpackEmptyContext.keys = () => ([]);\nwebpackEmptyContext.resolve = webpackEmptyContext;\nwebpackEmptyContext.id = `./src sync recursive`;\nmodule.exports = webpackEmptyContext;\n\n\n// sourceURL=webpack://webpack-demo./src/_sync?");
/***/ })
/******/ });
/*
*****/
/***/ // The module cache
/***/ var __webpack_module_cache__ = {};
/***/
/***/ // The require function
/***/ function __webpack_require__(moduleId) {
/***/ // Check if module is in cache
/***/ var cachedModule = __webpack_module_cache__[moduleId];
/***/ if (cachedModule !== undefined) {
/***/ return cachedModule.exports;
/***/ }
/***/ // Create a new module (and put it into the cache)
/***/ var module = __webpack_module_cache__[moduleId] = {
/***/ // no module.id needed
/***/ // no module.loaded needed
/***/ exports: {}
/***/ };
/***/
/***/ // Execute the module function
/***/ __webpack_modules__[moduleId](module, module.exports, __webpack_require__);
/***/
/***/ // Return the exports of the module
/***/ return module.exports;
/***/ }
/***/ }
/*
*****/
/***/ /* webpack/runtime/hasOwnProperty shorthand */
/***/ (O=> {
/***/ __webpack_require__.o = (obj, prop) => (Object.prototype.hasOwnProperty.call(obj, prop))
/***/ });
/*
*****/
/***/
/***/
/***/ // startup
/***/ // Load entry module and return exports
/***/ // This entry module can't be inlined because the eval devtool is used.
/***/ var __webpack_exports__ = __webpack_require__("./src/app.js");
/***/
/***/ })()
;
```

```
const config = {
  entry: "/src/app.js",
  output: {
    path: __dirname + "/dist",
    filename: "bundle.js",
  },
  mode: "development",
};
module.exports = config;
```

```
const priceEl = document.getElementById("price");
const balanceEl = document.getElementById("balance");
const expenseEl = document.getElementById("expense");
const expensesListEl = document.getElementById("expenses-list");
const submitBtn = document.getElementById("submit");
const resetBtn = document.getElementById("reset");
const subtract = require("calculation.js");

function addToList(name, price) {
    expensesListEl.innerHTML += `<li class="list-group-item">
        Name: ${name}
        <span class="ml-4">Price: ${price}</span></li>`;
}

function submit(e) {
    e.preventDefault();
    addToList(expenseEl.value, priceEl.value);
}

function reset(e) {
    e.preventDefault();
}

submitBtn.onclick = submit;
resetBtn.onclick = reset;
```

```
function subtract(a, b) {
  return a - b;
}
module.exports = {
  subtract,
};
```

```

const WebpackPwaManifest = require("webpack-pwa-manifest");
const path = require("path");
const config = {
  entry: "./public/assets/js/app.js",
  output: {
    path: __dirname + "/public/dist",
    filename: "bundle.js"
  },
  mode: "production",
  plugins: [
    new WebpackPwaManifest({
      // the name of the generated manifest file
      filename: "manifest.json",
      // we aren't using webpack to generate our html so we
      // set inject to false
      inject: false,
      // set fingerprints to `false` to make the names of the generated
      // files predictable making it easier to refer to them in our code
      fingerprints: false,
      name: "Images App",
      short_name: "Images App",
      theme_color: "#ffffff",
      background_color: "#ffffff",
      start_url: "/",
      display: "standalone",
      icons: [
        {
          src: path.resolve(
            __dirname,
            "public/assets/images/icons/icon-512x512.png"
          ),
          // the plugin will generate an image for each size
          // included in the size array
          size: [72, 96, 128, 144, 152, 192, 384, 512]
        }
      ]
    })
  ];
};
module.exports = config;

```

Plug-in pt1

```

const FILES_TO_CACHE = [
  '/',
  '/assets/css/style.css',
  '/index.html',
  '/favicon.ico',
  '/dist/manifest.json',
  '/dist/bundle.js',
  '/dist/icon_72x72.png',
  '/dist/icon_96x96.png',
  '/dist/icon_128x128.png',
  '/dist/icon_144x144.png',
  '/dist/icon_152x152.png',
  '/dist/icon_192x192.png',
  '/dist/icon_384x384.png',
  '/dist/icon_512x512.png',
  '/assets/images/1.jpg',
  '/assets/images/2.jpg',
  '/assets/images/3.jpg',
  '/assets/images/4.jpg',
  '/assets/images/5.jpg',
  '/assets/images/6.jpg',
  '/assets/images/7.jpg',
  '/assets/images/8.jpg',
  '/assets/images/9.jpg',
  '/assets/images/10.jpg',
  '/assets/images/11.jpg',
  '/assets/images/12.jpg',
  '/assets/images/13.jpg',
  '/assets/images/14.jpg',
  '/assets/images/15.jpg',
  '/assets/images/16.jpg',
  '/assets/images/17.jpg',
  '/assets/images/18.jpg',
  '/assets/images/19.jpg',
  '/assets/images/20.jpg',
  '/assets/images/21.jpg',
  '/assets/images/22.jpg',
  '/assets/images/23.jpg',
  '/assets/images/24.jpg',
  '/assets/images/25.jpg',
  '/assets/images/26.jpg',
  '/assets/images/27.jpg',
  '/assets/images/28.jpg',
  '/assets/images/29.jpg',
  '/assets/images/30.jpg',
  '/assets/images/31.jpg',
  '/assets/images/32.jpg',
  '/assets/images/33.jpg',
  '/assets/images/34.jpg',
  '/assets/images/35.jpg',
  '/assets/images/36.jpg',
  '/assets/images/37.jpg',
  '/assets/images/38.jpg'
];

```

```

const STATIC_CACHE = "static-cache-v1";
const RUNTIME_CACHE = "runtime-cache";

```

```

self.addEventListener("install", event => {
  event.waitUntil(
    caches
      .open(STATIC_CACHE)
      .then(cache => cache.addAll(FILES_TO_CACHE))
      .then(() => self.skipWaiting())
  );
});

```

// The activate handler takes care of cleaning up old caches.

```

self.addEventListener("activate", event => {
  const currentCaches = [STATIC_CACHE, RUNTIME_CACHE];
  event.waitUntil(
    caches
      .keys()
      .then(cacheNames => {
        // return array of cache names that are old to delete
        return cacheNames.filter(
          cacheName => !currentCaches.includes(cacheName)
        );
      })
      .then(cachesToDelete => {
        return Promise.all(
          cachesToDelete.map(cacheToDelete => {
            return caches.delete(cacheToDelete);
          })
        );
      })
      .then(() => self.clients.claim())
  );
});

```

```

self.addEventListener("fetch", event => {
  // non GET requests are not cached and requests to other origins are not cached
  if (
    event.request.method !== "GET" ||
    !event.request.url.startsWith(self.location.origin)
  ) {
    event.respondWith(fetch(event.request));
    return;
  }
  // handle runtime GET requests for data from /api routes
  if (event.request.url.includes("/api/images")) {
    // make network request and fallback to cache if network request fails (offline)
    event.respondWith(
      caches.open(RUNTIME_CACHE).then(cache => {
        return fetch(event.request)
          .then(response => {
            cache.put(event.request, response.clone());
            return response;
          })
          .catch(() => caches.match(event.request));
      })
    );
    return;
  }
}

```

// use cache first for all other requests for performance

```

event.respondWith(
  caches.match(event.request).then(cachedResponse => {
    if (cachedResponse) {
      return cachedResponse;
    }
    // request is not in cache. make network request and cache the response
    return caches.open(RUNTIME_CACHE).then(cache => {
      return fetch(event.request).then(response => {
        return cache.put(event.request, response.clone()).then(() => {
          return response;
        });
      });
    });
  });

```

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <meta http-equiv="X-UA-Compatible" content="ie=edge" />
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/
    bootstrap/4.3.1/material/bootstrap.css" />
  <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.8.2/
    css/all.css"
    integrity="sha384-oS3vJWv+0UjzBfQzYUhtDYW+
    Pj2yciDJxpsK1OYPAYjqT085Qq/1cq5FLXAZQ7Ay"
    crossorigin="anonymous" />
  <link rel="stylesheet" href="/assets/css/style.css" type="text/css" />

  <!-- Changed to use the manifest generated by the webpack plugin -->
  <link rel="manifest" href="dist/manifest.json" />

  <title>Image Gallery</title>
</head>
<body>
  <nav class="navbar navbar-dark bg-primary">
    <a class="navbar-brand" href="#">Image Gallery</a>
  </nav>
  <div class="container"></div>

  <!-- Changed to use the output bundle generated by webpack -->
  <script src="/dist/bundle.js"></script>

  <script>
    if ("serviceWorker" in navigator) {
      window.addEventListener("load", () => {
        navigator.serviceWorker.register("service-worker.js").then(reg => {
          console.log("We found your service worker file!", reg);
        });
      });
    }
  </script>
</body>
</html>

```

Plug In pt2

```

{
  "name": "images-demo",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "webpack": "webpack",
    "seed": "node seeders/seed.js"
  },
  "author": "",
  "license": "ISC",
  "dependencies": {
    "express": "^4.16.4",
    "glob": "^7.1.4",
    "mongoose": "^5.5.8"
  },
  "devDependencies": {
    "webpack": "^4.32.0",
    "webpack-cli": "^3.3.2"
  }
}

```

```

var db = require("../models");

module.exports = function(app) {
  app.get("/api/images", function(req, res) {
    db.Image.find({}).then(function(dbImages) {
      res.json(dbImages);
    });
  });

  app.put("/api/images/:id", function(req, res) {
    db.Image.updateOne(
      { _id: req.params.id },
      { rating: req.body.rating }
    ).then(function(dbImage) {
      res.json(dbImage);
    });
  });
};

```

```

var express = require("express");
var mongoose = require("mongoose");

```

```

var app = express();
var PORT = process.env.PORT || 3000;

```

```

app.use(express.static("public"));
app.use(express.urlencoded({ extended: true }));
app.use(express.json());

```

```

require("./routes/apiRoutes")(app);
require("./routes/htmlRoutes")(app);

```

```

mongoose.connect(
  process.env.MONGODB_URI || "mongodb://localhost/
  webpackplugin",
  { useNewUrlParser: true, useUnifiedTopology: true }
);

```

```

app.listen(PORT, function() {
  console.log(`Now listening on port: ${PORT}`);
});

```

```

var path = require("path");

```

```

module.exports = function(app) {
  app.get("*", function(req, res) {
    res.sendFile(path.join(__dirname, "../public/index.html"));
  });
};

```

Pure Functions

```
import { subtract } from "./calculations";
import { balanceEl, expenseEl, expensesListEl, expensesChart, priceEl, submitBtn, resetBtn } from "./elements";
import { updateChart, resetChart } from "./expenseChart";
```

```
function addToList(name, price) {
  expensesListEl.innerHTML += `<li class="list-group-item">Name: ${name}
    <span class="ml-4">Price: ${price}</span></li>`;
}
```

```
function submit(e) {
  e.preventDefault();
  const total = subtract(Number(balanceEl.innerText), priceEl.value);
  balanceEl.innerText = total;
  addToList(expenseEl.value, priceEl.value);
  updateChart(expensesChart, expenseEl.value, priceEl.value);
}
```

```
function reset(e) {
  e.preventDefault();
  const total = 2000;
  balanceEl.innerText = total;
  expensesListEl.innerHTML = "";
  resetChart(expensesChart);
}
submitBtn.onclick = submit;
resetBtn.onclick = reset;
```

```
export function updateChart(expenseChart, name, price) {
  expenseChart.data.labels.push(name);
  expenseChart.data.datasets.forEach((dataset) => {
    dataset.data.push(price);
  });
  expenseChart.update();
}
```

```
export function resetChart(expenseChart) {
  expenseChart.data.labels = [];
  expenseChart.data.datasets.forEach((dataset) => {
    dataset.data = [];
  });
  expenseChart.update();
}
```

```
export function subtract(a, b) {
  const result = a - b;
  return result;
}
```

```
const Chart = require("chart.js");

export const priceEl = document.getElementById("price");

export const balanceEl = document.getElementById("balance");

export const expenseEl = document.getElementById("expense");

export const expensesListEl = document.getElementById(
  "expenses-list");

export const submitBtn = document.getElementById("submit");

export const resetBtn = document.getElementById("reset");

const ctx = document.getElementById("myChart").getContext(
  "2d");

export const expensesChart = new Chart(ctx, {
  type: "bar",
  data: {
    labels: [],
    datasets: [{
      label: "Expense cost in $",
      data: [],
      backgroundColor: [
        "rgba(255, 99, 132, 0.2)",
        "rgba(54, 162, 235, 0.2)",
        "rgba(255, 206, 86, 0.2)",
        "rgba(75, 192, 192, 0.2)",
        "rgba(153, 102, 255, 0.2)",
        "rgba(255, 159, 64, 0.2)"
      ],
      borderColor: [
        "rgba(255, 99, 132, 1)",
        "rgba(54, 162, 235, 1)",
        "rgba(255, 206, 86, 1)",
        "rgba(75, 192, 192, 1)",
        "rgba(153, 102, 255, 1)",
        "rgba(255, 159, 64, 1)"
      ],
      borderWidth: 1
    }]
  },
  options: {
    title: {
      text: "Expenses Chart"
    },
    scales: {
      yAxes: [{
        ticks: {
          beginAtZero: true
        }
      }]
    }
  }
});
```

Chunking

- > models
- ▼ public
 - ▼ assets
 - > css
 - > images
 - ▼ js
 - JS api.js
 - JS cardCreation.js
 - JS detail.js
 - JS domMethods.js
 - JS favorites.js
 - JS home.js
 - JS indexedDb.js
 - JS rating.js
 - <> detail.html
 - ★ favicon.ico
 - <> favorites.html
 - <> index.html
 - JS service-worker.js
- ▼ routes
 - JS apiRoutes.js
 - JS htmlRoutes.js
- ▼ seeders
 - JS seed.js
- ◆ .gitignore
- { } package.json
- JS server.js
- 🔌 webpack.config.js

```
import { loadImages } from "../api";
import { createCards } from "../cardCreation";
```

```
loadImages().then(data => createCards(data));
```

```
import { useIndexedDb } from "../indexedDb";
import { createCards } from "../cardCreation";
```

```
useIndexedDb("favorites", "FavoritesStore", "get").then(results => {
  createCards(results);
});
```

```
const WebpackPwaManifest = require("webpack-pwa-manifest");
const path = require("path");
```

```
const config = {
  mode: "production",
  entry: {
    app: "../public/assets/js/home.js",
    detail: "../public/assets/js/detail.js",
    favorites: "../public/assets/js/favorites.js"
  },
  output: {
    path: __dirname + "/public/dist",
    filename: "[name].bundle.js"
  },
  plugins: [
    new WebpackPwaManifest({
      fingerprints: false,
      inject: false,
      name: "Images App",
      short_name: "Images App",
      description: "An application for images",
      background_color: "#01579b",
      theme_color: "#ffffff",
      start_url: "/",
      icons: [{
        src: path.resolve("public/assets/images/icons/icon-192x192.png"),
        sizes: [96, 128, 192, 256, 384, 512],
        destination: path.join("assets", "icons")
      }]
    })
  ],
  module: {
    rules: [
      {
        test: /\.m?js$/,
        exclude: /(node_modules)/,
        use: {
          loader: "babel-loader",
          options: {
            presets: ["@babel/preset-env"]
          }
        }
      }
    ]
  }
};
```

```
module.exports = config;
```