

Packages Installed

1. Node modules

npm init

2. Nodemon

npm install --save-dev nodemon

3. Express

npm install --save express

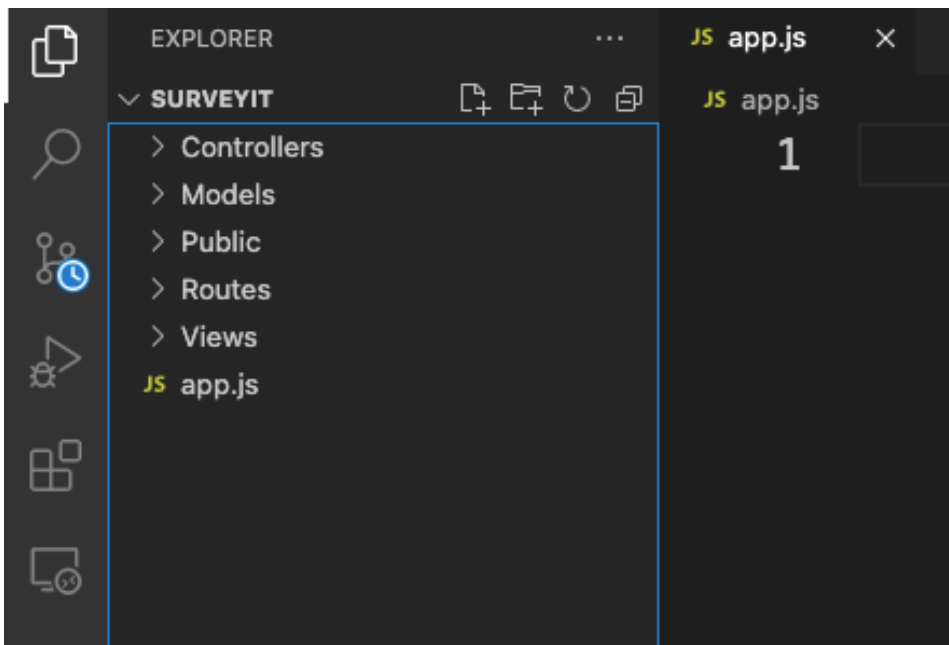
4. Ejs

npm install --save ejs

5. Body Parser

npm install --save body-parser

Step 1: Preparing My Workspace



Creating the app.js, MVC, Routes and Public Folder

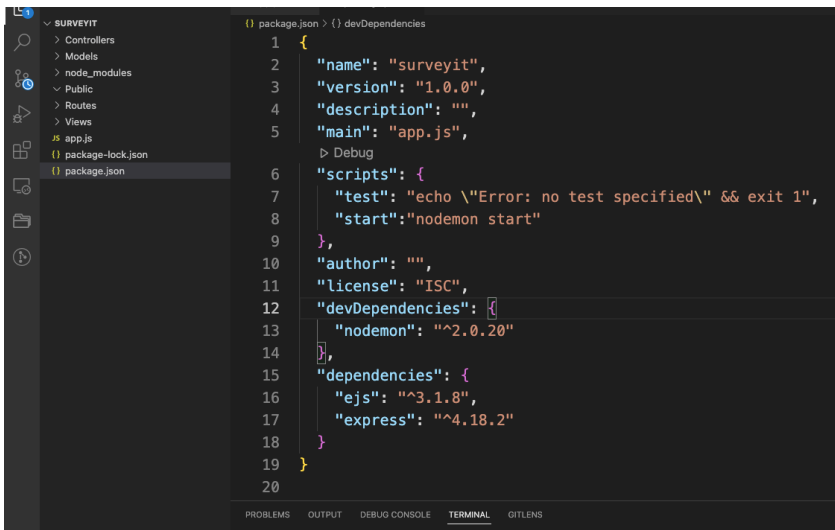
Installing the packages

- Nodemon
- Express
- Nodemon

Editing scripts in package.json

Adding Start Key

```
"scripts": {  
  "test": "echo \"Error: no test specified\" &&  
  exit 1",
```

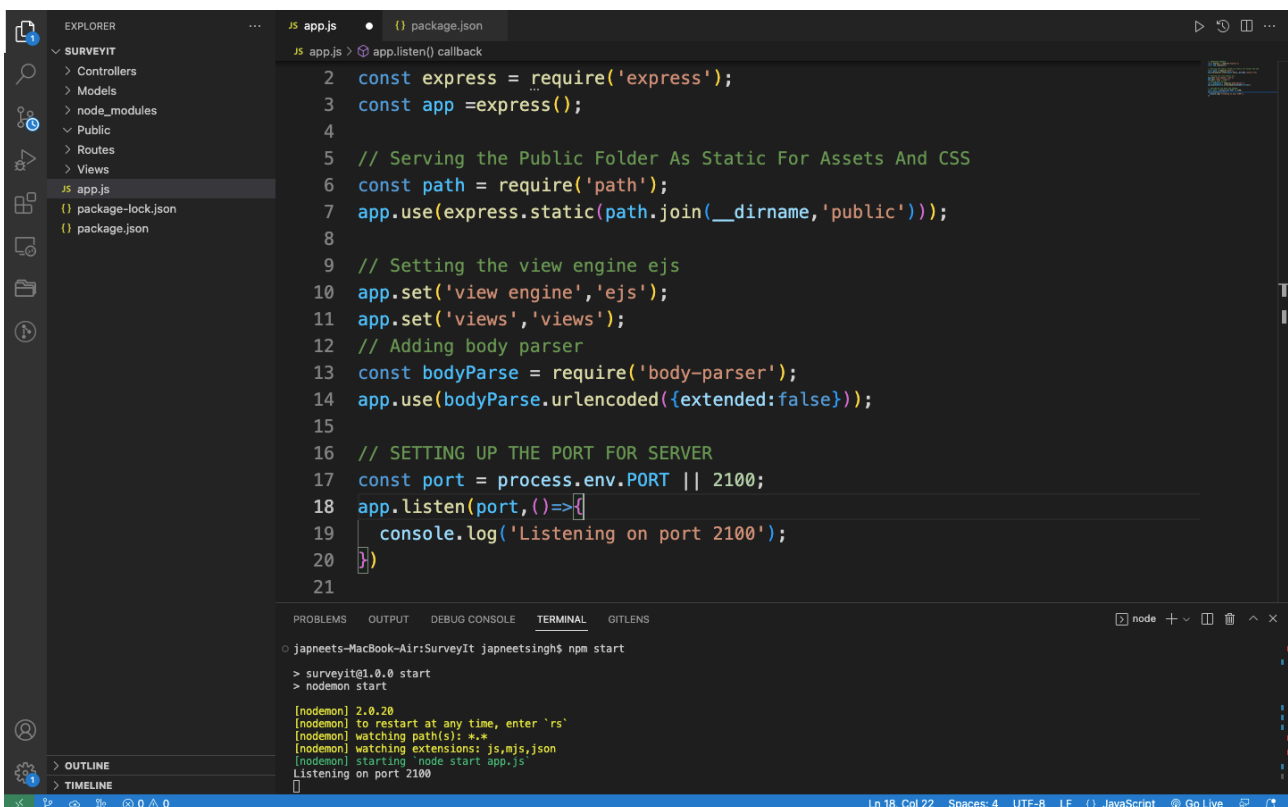


```
1 {
2   "name": "surveyit",
3   "version": "1.0.0",
4   "description": "",
5   "main": "app.js",
6   "scripts": {
7     "test": "echo \\\"Error: no test specified\\\" && exit 1",
8     "start": "nodemon start"
9   },
10  "author": "",
11  "license": "ISC",
12  "devDependencies": {
13    "nodemon": "^2.0.20"
14  },
15  "dependencies": {
16    "ejs": "^3.1.8",
17    "express": "^4.18.2"
18  }
19 }
```

`"start": "nodemon start"`
`},`

Nodemon start will start our **nodemon**

Setting up app.js



```
1 // app.js
2 const express = require('express');
3 const app = express();
4
5 // Serving the Public Folder As Static For Assets And CSS
6 const path = require('path');
7 app.use(express.static(path.join(__dirname, 'public')));
8
9 // Setting the view engine ejs
10 app.set('view engine', 'ejs');
11 app.set('views', 'views');
12 // Adding body parser
13 const bodyParser = require('body-parser');
14 app.use(bodyParser.urlencoded({extended: false}));
15
16 // SETTING UP THE PORT FOR SERVER
17 const port = process.env.PORT || 2100;
18 app.listen(port, () => {
19   console.log('Listening on port 2100');
20 });
21
```

Terminal Output:

```
japneets-MacBook-Air:SurveyIt japneetsingh$ npm start
> surveyit@1.0.0 start
> nodemon start

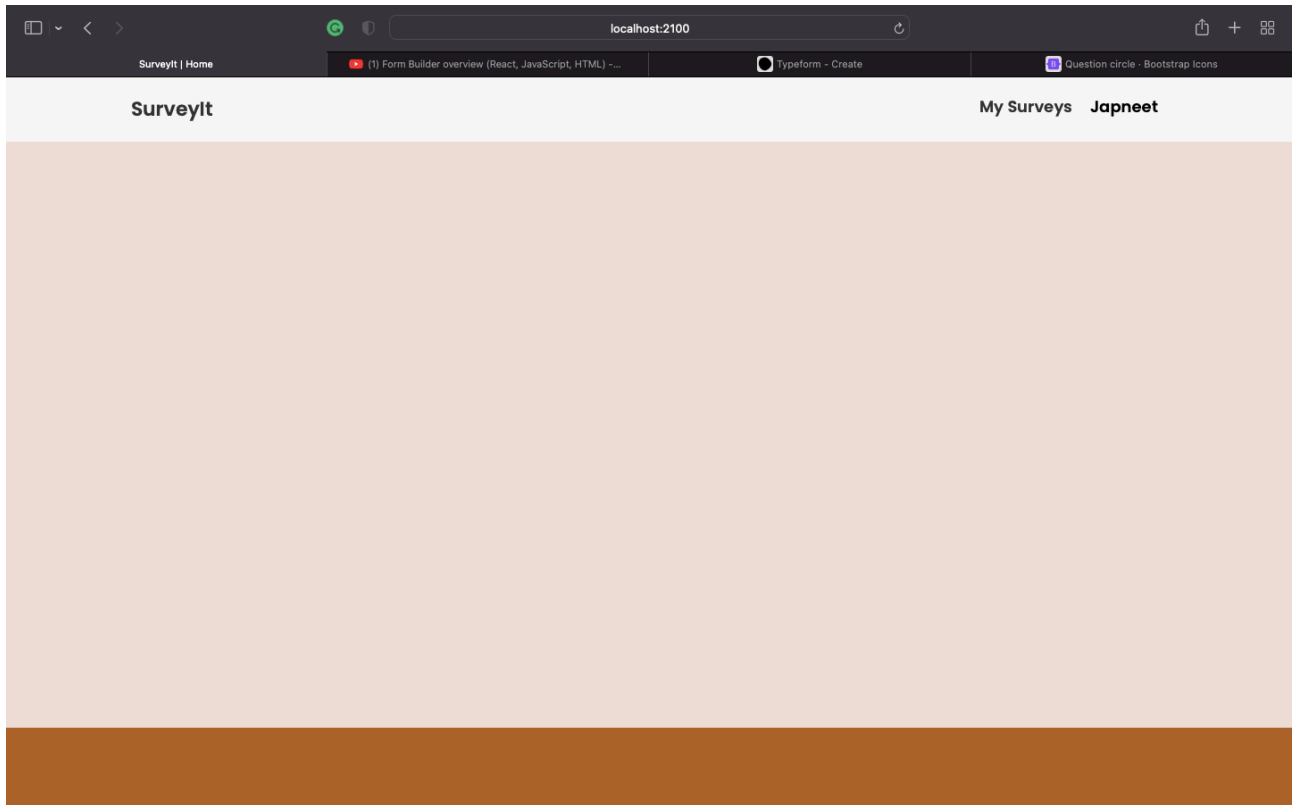
[nodemon] 2.0.20
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node start app.js`
Listening on port 2100
```

Starting the server using **npm start**

STEP 2: SETTING UP THE FRONTEND

For frontend I have used HTML, CSS and Javascript. Template Engine used is EJS

Setting up the routes file to load pages



First of all I have created landing page of our application which renders on '/'

Then I worked on My Surveys Section You Can see in the navbar This will only be visible to the user until he is logged in

Then I prepared the basic structure of all pages which are kept in dashboard-includes folder

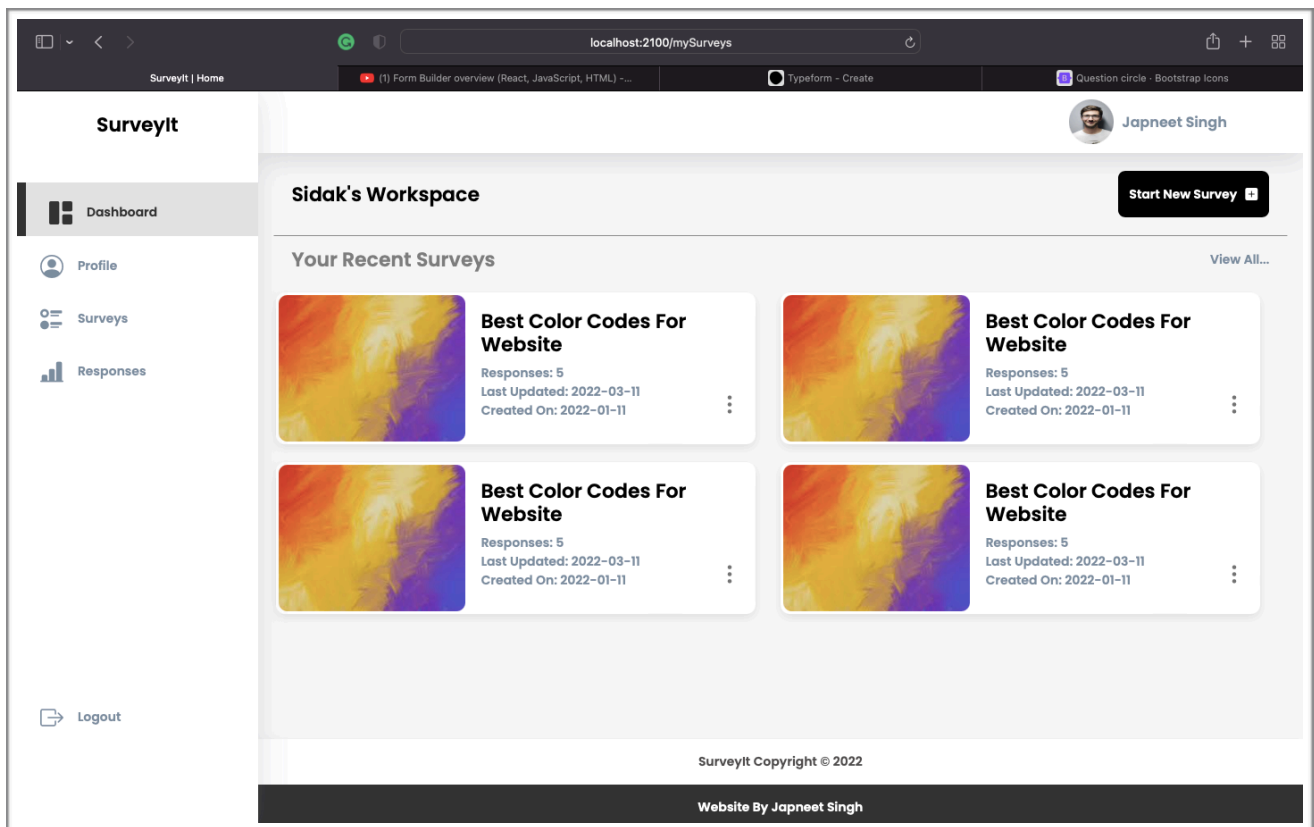
1 Sidebar

2 Footer

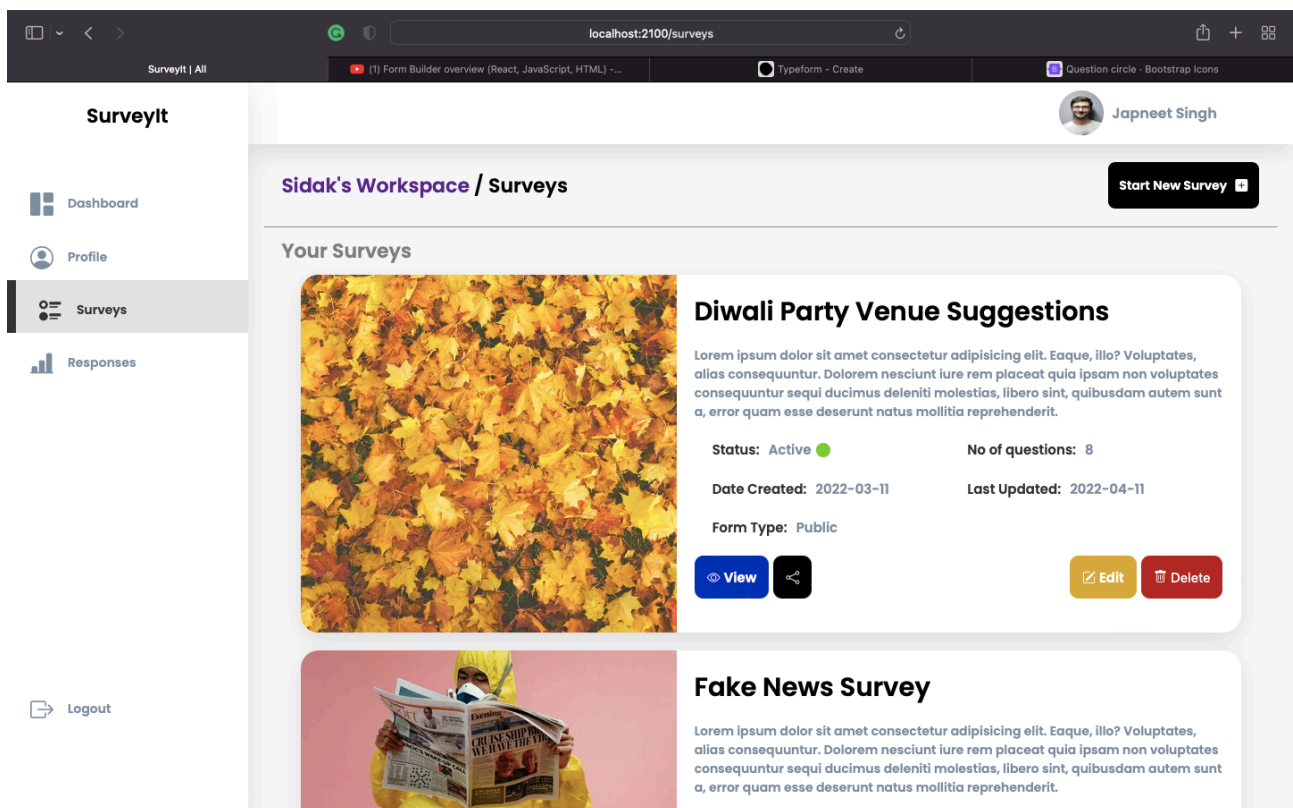
3 Header

After designing the main components I designed my dashboard page that displays last 4 surveys created

Will work on it later



After dashboard section I coded the front end for surveys page which displays all the surveys and their details with edit, view, share, delete functionality



After the surveys page I started designing the frontend for create new survey page

