Project Plan (Zionet Project – Lists)

What technologies will be used in this project?

MERN – MongoDB, ExpressJS, ReactJS, NodeJS.

What are the requirements of the project?

1. Authentication – connect to the application with a personal user, either through SSO or through username (email) & password.
   1. Register button:
      1. “Continue with SSO” (Google)
      2. Regular register (enter email-address and password).
   2. Login (first screen of the webpage):

What technologies will be used to develop the front-end of the project?

1. ExpressJS.

What technologies will be used to develop the back-end of the project?

1. NodeJS and ExpressJS (ExpressJS is a framework that’s used with NodeJS to create the server and allow routing).
2. MongoDB – to create a database.

Pages (the pages of the application):

1. Login page (main page):
   1. Allows the user to login with a registered user.
   2. Allows the user go to the register page in order to register.
2. Register page:
3. About page:
4. Lists page (preview page):
5. Lists page (edit page):

What details do we need to store in the database?

1. User credentials:
   1. SSO (Google) – Google email and password to Google account. (the ID should be saved)
   2. Regular – specified email and specified password.

{

“\_id”: 0,

“provider”: 1, (0 – none-SSO / 1 – Google / 2 – Facebook)

“username”: “something@yahoo.com”,

“password”: 12345,

“safeWord”: jsei5u30945u,

“ssoId”: 89123u891239,

“date-created”: <date>,

“settings”:

{

“view-mode”: <1 – light / 0 – dark>,

}

}

{

“\_id”: 950,

“provider”: 0,

“username”: “sometone@gmail.com”,

“password”: U\*(U#$\*(U\*(34,

“safeWord”: \*WU\*(EU\*(#$\*(#U$\*(

}

{

“\_id”: 1003,

“provider”: 1,

“ssoId”: d23894589234789234

}

1. List details:
   1. The email of the owner of the list.
   2. The general list:
      1. Priority of the list (1 – 10, 1 being the least, 10 being the most).
      2. Shares & permissions (the email address of the user whom we shared the list with, along with permissions: 1=read / 2=read + write).
      3. Date created.
   3. Header of the list:
      1. The string of the list (example: “Shopping List”).
      2. The color of the list (example: “rgb(100,100,100)”).
   4. Tasks of the list:
      1. The string of the task (example: “apples”).
      2. Is crossed-out (false – not crossed-out / true – is crossed-out).
      3. Is deleted (false – not deleted / true – is deleted).
      4. Does contains sub-tasks? (false – doesn’t contain / true – does contain).

{

“\_id”: <id of the list (starts from 0 – infinity),

“priority”: <number 1-10>,

“owner”: <id of owner>,

“date-created”: <timestamp>,

“date-modified”: <timestamp>,

“color”: <color>,

“header”:

{

“string”: <name of list>,

“color”: <name of color>,

}

“tasks”:

{

“\_id”: <id>,

“string”: <name of task>,

“sub-tasks”:

{

“\_id”: <id of list of sub tasks>,

{

“\_id of sub tasks”: [7,8,9],

{

“\_id of sub task”: <id>,

“name of sub task”: <name”,

“isCrossedOut”: <true / false>,

“isDeleted”: <true / false>,

}

}

}

}

Document of collaboration:

{

“\_id”: <starts from zero>

“listID”:<id of the list>

“users”:

{

{

“\_id”:number,

“userID”:id,

“premmisons”:<1 – read only / 2 – read + write>

},

{

“\_id”:number,

“userID”:id,

“premmisons”:<1 – read only / 2 – read + write>

},

}

}

1. Settings:
   1. [OPTIONAL] Dark mode. False = disabled (normal light mode) / True = enabled (dark mode).
   2. We need to support MVP
   3. <date of user creation>

Lists

1. Preview Mode / Edit Mode (this is NOT the permissions that are given to the user, during collaboration):
   1. Preview Mode – this is where we view a preview of all the lists that have been created by the user / that have been shared with the user. Each list will look like a card, and only a few characters can be seen in each card.
   2. Edit Mode – this is where we view the list in full size, ready to be read and / or edited. The other lists that have not been selected will appear blur.
2. Headers – can only change the color of the headers, cannot change the font-size / font-family of the headers.
   1. When viewing headers in preview-mode, only the first X characters will be displayed. After the X characters, will come “…” (to indicate the continuance of the string).
   2. When viewing the headers in edit-mode, the entire header should get displayed onto the screen.
3. Colors – there will be a row of light colors, on top of the row of dark colors:
   1. Light & dark colors:

|  |  |  |
| --- | --- | --- |
| Color | Light (hexadecimal value) | Dark (hexadecimal value) |
| Green | Rgb(160,255,147) | Rgb(13,98,0) |
| Blue | Rgb(147,239,255) | Rgb(0,116,136) |
| Yellow | Rgb(255,255,147) | Rgb(102,102,0) |
| Red | Rgb(255,147,147) | Rgb(130,0,0) |
| Orange | Rgb(255,211,147) | Rgb(153,91,0) |
| Purple | Rgb(185,147,255) | Rgb(56,0,159) |
| Light-grey (default light mode) | Rgb(255,255,255) | Null |
| Dark-grey (default dark mode) | Null | Rgb(65,65,65) |

1. Sub-tasks (“Lesser Task” – there will be a possibility to add a “sub-task” to each of the existing tasks. The sub-tasks will be a list, which will only be visible once clicking on the “greater-task” (the task from which the sub-tasks are created). Some terminology:
   1. “Task” – the name of the task from which we create the sub-tasks.
   2. “Sub task” – the name of the task that stems from the main-task.
2. Share – to invite collaborators to either read / read + write:
   1. “Restricted” – allows the user to read the list only.
   2. “Advanced” – allows the user to read and edit the lists.