

# Planify

## Python Django REST Framework Website

# Devlog



# 21st September 2022 (21/09/2022)

The screenshot shows a Windows desktop environment. On the left, a Microsoft Edge browser window is open at the URL `127.0.0.1:8000`, displaying a simple web application titled "Planify". The application has a header "Project 1 Week 2" and a table with two rows: one for "Homework" (Green, Green) and one for "Exam" (Red, Green). On the right, a Visual Studio Code (VS Code) interface is visible. The code editor shows the file `home.html` with the following content:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Planify</title>
    <style>
    </style>
  </head>
  <body>
    <h1>Planify</h1>
    <div id="table-list">
      <table>
        <tr>
          <th>Project 1</th>
          <th>Week 1</th>
          <th>Week 2</th>
        </tr>
        <tr>
          <td>Homework</td>
          <td>Green</td>
          <td>Green</td>
        </tr>
        <tr>
          <td>Exam</td>
          <td>Red</td>
          <td>Green</td>
        </tr>
      </table>
    </div>
  </body>
<script>
  function getCookie(name) {
    var cookieValue = null;
    if (document.cookie && document.cookie !== '') {
      var cookies = document.cookie.split(';');
      for (var i = 0; i < cookies.length; i++) {
        var cookie = cookies[i].trim();
        // Does this cookie string begin with the name we want?
        if (cookie.substring(0, name.length + 1) === (name + '=')) {
          cookieValue = decodeURIComponent(cookie.substring(name.length + 1));
          break;
        }
      }
    }
  }
</script>
```

The VS Code interface includes tabs for `viewsp.py`, `models.py`, `serializers.py`, `urls.py`, and `home.html`. Below the code editor, the terminal shows the output of a Django development server:

```
Watching for file changes with StatReloader
Performing system checks...
System check identified no issues (0 silenced).
September 21, 2022 - 09:57:30
Django version 4.1.1, using settings 'planify.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
[21/Sep/2022 09:57:32] "GET / HTTP/1.1" 200 1429
[21/Sep/2022 09:57:54] "GET / HTTP/1.1" 200 1515
```

The bottom status bar of the VS Code window displays: Ln 27, Col 28, Spaces: 4, UTRF-8, CRLF, HTML, Go Live, and the date/time: 09:58 21/09/2022.

# 21st September 2022 (21/09/2022)

The screenshot shows a developer's workspace with multiple windows open:

- Browser Window:** Displays a simple web application titled "Planify". The page content includes:
  - Project 1 Week 1 Week 2**
  - Homework Green Green
  - Exam Red Green
- Code Editor:** Visual Studio Code window showing the file `home.html`. The code contains JavaScript logic to handle asynchronous requests and generate HTML tables based on data from `base.html`.
- Terminal:** Shows the command `python` running in the background.
- Network Tab:** Shows a request to `GET http://127.0.0.1:8000/favicon.ico` with a status of 404 Not Found.
- Output Tab:** Displays log entries from the application, including several requests to `/getTable` with status codes 200 and 202.

# 21st September 2022 (21/09/2022)

The screenshot shows a developer's workspace with multiple windows open:

- Browser Window:** Displays a local development site titled "Planify". The page content includes a heading "Project 1 Week 1 Week 2" and a list of items: Exam, Homework, Sleep, and Fail.
- Code Editor:** An instance of Visual Studio Code is open, showing the file `home.html`. The code is a JavaScript file that generates HTML tables based on data from `base.html` and `displayTables()`.
- Inspector Tools:** A DevTools panel is visible, specifically the "Elements" tab under "Inspector". It shows the DOM structure of the page, with the `<body>` element selected. The "Changes" tab is active, displaying the message "No changes found." and noting that changes to CSS will appear here.
- Terminal:** The bottom right corner shows a terminal window with a log of HTTP requests and responses, indicating successful fetches for various API endpoints like `/getSubjects/1`, `/getWeeks/1`, and `/getTable/1`.

# 21st September 2022 (21/09/2022)

The screenshot shows a developer's workspace with multiple windows open:

- Browser Window:** Displays a simple web application titled "Planify". The page content includes:
  - Project 1 Week 1 Week 2**
  - Exam false
  - Homework false
  - Sleep
  - Fail true
- Code Editor:** Visual Studio Code is open, showing the file `views.py`. The code contains Python and JavaScript混用的逻辑，用于生成HTML表格。关键部分如下：

```
function displayTables()
{
    var table_list = document.getElementById('table-list')
    var item =
        `<table>
        <tr>
        <th>${tab_data[0].name}</th>
        for (var i = 0; i < week_data.length; i++)
        {
            item += `<th>Week ${week_data[i].number}</th>
        }
        item += `</tr>

        //console.log(node_data)
        for (var i = 0; i < subject_data.length; i++)
        {
            item += `<tr>
            item += `<td>${subject_data[i].name}</td>
            for (var j = 0; j < node_data.length; j++)
            {
                if (node_data[j].subject == subject_data[i].id)
                    item += `<td>${node_data[j].completed}</td>
            }
            item += `</tr>
        }

        item += `</table>
        table_list.innerHTML += item
    }
}

function getTable(tableSpecifier)
{
    var url = 'http://127.0.0.1:8000/getTable/${tableSpecifier}/'

    fetch(url)
    .then((resp) => resp.json())
    .then(function(data){
        tab_data = data
    })
}
```
- Terminal:** Shows a log of HTTP requests from the application to the server at `127.0.0.1:8000`. The log includes:
  - 2022-09-21 15:19:48 "GET /favicon.ico" 200 0ms
  - 2022-09-21 15:19:48 "GET /getTable/1/ HTTP/1.1" 200 29
  - 2022-09-21 15:19:48 "GET /getWeeks/1/ HTTP/1.1" 200 61
  - 2022-09-21 15:19:48 "GET /getSubjects/1/ HTTP/1.1" 200 138
  - 2022-09-21 15:19:48 "GET /getNodes/1/ HTTP/1.1" 200 144
  - 2022-09-21 15:20:07 "GET / HTTP/1.1" 200 3771
  - 2022-09-21 15:20:07 "GET /getTable/1/ HTTP/1.1" 200 29
  - 2022-09-21 15:20:07 "GET /getSubjects/1/ HTTP/1.1" 200 138
  - 2022-09-21 15:20:07 "GET /getWeeks/1/ HTTP/1.1" 200 61
  - 2022-09-21 15:20:07 "GET /getNodes/1/ HTTP/1.1" 200 144

# 21st September 2022 (21/09/2022)

The screenshot shows a developer's environment with multiple windows open:

- Browser:** A Microsoft Edge window displaying a local development site at `127.0.0.1:8000`. The page title is "Planify". It lists a table of data:

	Project 1	Week 1	Week 2	Week 13
Exam	false			
Homework	false	true		
Sleep	true	false		
Fail	true			
- Code Editor:** A Visual Studio Code window showing the file `views.py`. The code implements several API endpoints using Django's REST framework:

```
base = views.BaseView.as_view()  
@api_view(['GET'])  
def getSubjects(request, pk):  
    table = Table.objects.all().filter(id=pk)  
    subjects = table[0].subjectToTable.all()  
    subjectsSerializer = SubjectSerializer(subjects, many=True)  
    return Response(subjectsSerializer.data)  
  
@api_view(['GET'])  
def getWeeks(request, pk):  
    table = Table.objects.all().filter(id=pk)  
    weeks = table[0].weekToTable.all()  
    weeksSerializer = WeekSerializer(weeks, many=True)  
    return Response(weeksSerializer.data)  
  
@api_view(['GET'])  
def getNodes(request, pk):  
    table = Table.objects.all().filter(id=pk)  
    subjects = table[0].subjectToTable.all()  
    #print(f"Subjects: {len(subjects)}")  
    #print(subjects[0])  
    nodes = []  
    for subject in subjects:  
        if subject != None:  
            nodes.append(subject.nodeSubject.filter(subject=subject))  
    #print(nodes)  
    n = []  
    for i in range(0, len(nodes)):  
        #print(f"i={i}")  
        for j in range(0, len(nodes)):  
            try:  
                n.append(nodes[i][j])  
            except IndexError:  
                continue  
    #print(f"item: {nodes[i]}")  
    #print(n)  
    nodeSerializer = NodeSerializer(n, many=True)  
    return Response(nodeSerializer.data)  
    #return Response(None)  
  
#TODO:  
# code works so far, returns desired nodes. however these are encapsulated in querysets which make it  
# impossible to serialize. extract them
```
- Terminal:** A terminal window showing log output from a Python application. The logs include requests to the application's API and errors related to a broken pipe:

```
[21/Sep/2022 15:38:40,059] - Broken pipe from ('127.0.0.1', 53111)  
[21/Sep/2022 15:38:40,061] - Broken pipe from ('127.0.0.1', 53108)  
[21/Sep/2022 15:38:45] "POST /admin/base/node/add/ HTTP/1.1" 302 0  
[21/Sep/2022 15:38:45] "GET /admin/base/node/ HTTP/1.1" 200 8927  
[21/Sep/2022 15:38:45] "GET /admin/jstl1n/ HTTP/1.1" 200 3343  
[21/Sep/2022 15:38:46] "GET / HTTP/1.1" 200 3668  
[21/Sep/2022 15:38:47] "GET /getTable/1/ HTTP/1.1" 200 29  
[21/Sep/2022 15:38:47] "GET /getSubjects/1/ HTTP/1.1" 200 138  
[21/Sep/2022 15:38:47] "GET /getWeeks/1/ HTTP/1.1" 200 92  
[21/Sep/2022 15:38:47] "GET /getNodes/1/ HTTP/1.1" 200 286
```
- Bottom Taskbar:** Shows various system icons including the Start button, File Explorer, Task View, Taskbar settings, and a clock indicating 15:39 on 21/09/2022.

# 23rd September 2022 (23/09/2022)

The screenshot shows a dual-pane development environment. On the left is a web browser window displaying a local application at 127.0.0.1:8000. The page title is "Planify" and it lists a table of project tasks:

Project 1 Week 1 Week 2		
Exam	false	false
Homework	false	false
Sleep	true	false
Fail	false	true
Revise	false	false
Abandon	false	false

On the right is a Visual Studio Code editor window. The left sidebar shows a file tree for a Django-like project structure under "PLANIFY". The main editor pane displays a Python file named "views.py" containing code for handling API requests. The code includes logic for creating subjects, adding nodes to subjects, and managing weeks. The terminal tab at the bottom shows a log of HTTP requests and responses.

```
base = views.py - planify - Visual Studio Code
File Edit Selection View Go Run Terminal Help
home.html models.py views.py uris.py
base
  base
    > __pycache__
    > migrations
    > templates
      base
        > home.html
        > __init__.py
        > admin.py
        > apps.py
        > models.py
        > serializers.py
        > tests.py
        > urls.py
        > views.py
      planify
      db.sqlite3
      manage.py

@api_view(['POST'])
def createSubject(request, pk):
    # get the table
    # create the subject
    # for each week add a node to link subject and week with node
    table = Table.objects.all().filter(id=pk)
    #print("table set")
    subject = SubjectSerializer(data=request.data)
    #print("subject serialized")
    if subject.is_valid():
        subject.save()
        #print("subject saved to db")
    weeks = table[0].weekToTable.all()
    #print("obtained weeks")
    for week in weeks:
        #print(week.id)
        #print("iterating weeks")
        #d = {"completed": False, "subject": subject.data['id'], "week": week.id}
        #d = {"completed": False, "subject": subject.data['id'], "week": week.id}
        #print(f"data = {data}")
        #print(request.data)
        #print(d)
        data = {
            "subject": subject.data['id'],
            "week": week.id
        }
        node = NodeSerializer(data=data)
        #print(node.initial_data[0])
        if node.is_valid():
            #print("node is valid")
            node.save()
    return Response(None)

@api_view(['POST'])
def testApi(request):
    [23/Sep/2022 10:11:09] "GET /getSubjects/1/ HTTP/1.1" 200 211
    [23/Sep/2022 10:11:09] "GET /getNodes/1/ HTTP/1.1" 200 585
    [23/Sep/2022 10:11:15] "GET /admin/base/node/ HTTP/1.1" 200 10055
    [23/Sep/2022 10:11:15] "GET /admin/jis18n/ HTTP/1.1" 200 3343
    [23/Sep/2022 10:11:16] "GET /admin/base/node/23/change/ HTTP/1.1" 200 10291
    [23/Sep/2022 10:11:16] "GET /admin/base/node/ HTTP/1.1" 200 10055
    [23/Sep/2022 10:11:17] "GET /admin/base/node/22/change/ HTTP/1.1" 200 10291
    [23/Sep/2022 10:11:17] "GET /admin/jis18n/ HTTP/1.1" 200 3343
    [23/Sep/2022 10:11:18] "GET /admin/base/node/ HTTP/1.1" 200 10055
```

# 23rd September 2022 (23/09/2022)

The screenshot shows a Windows desktop environment. On the left, a Microsoft Edge browser window displays a local development site at `127.0.0.1:8000`. The page title is "Planify" and it shows a table titled "Project 1 Week 1 Week 2" with the following data:

	Exam	Homework	Sleep	Fail	Revise
Exam	false	false			
Homework	false	false			
Sleep	true	false			
Fail	false	true			
Revise	false	false			

On the right, a Visual Studio Code (VS Code) window is open, showing the code for `home.html` under the `base/templates/base` directory. The code generates HTML for a table based on data from `complete_table`:// Construct HTML representation of above table  
var item = `<table>  
//console.log(complete\_table)  
  
// Table and weeks  
item += `<tr>  
item += `<th>\${complete\_table[0][0]}</th>  
for (var i = 1; i < complete\_table[0].length; i++)  
{  
 item += `<th>Week \${complete\_table[0][i].number}</th>  
}  
item += `</tr>  
  
// Subjects and nodes  
for (var sIter = 1; sIter < complete\_table.length; sIter++)  
{  
 item += `<tr>  
 item += `<td>\${complete\_table[sIter][0]}</td>  
 // Nodes  
 for (var nIter = 1; nIter < complete\_table[0].length; nIter++)  
 {  
 item += `<td><button>\${complete\_table[sIter][nIter]}</button></td>  
 }  
 item += `</tr>  
}  
  
item += `</table>  
table\_list.innerHTML += item  
}  
  
// delete functionality:  
// - whilst iterating in above code, match the representation to actual object  
// - grab the class names attached and add event listeners  
// - for this to happen you need to already add the items to innerHTML  
  
// idea:  
// - first iterate weeks whilst adding listeners  
// - then iterate subjects adding listeners  
// - finally iterate nodes adding listeners  
  
function getTable(tableSpecifier)  
{  
 var url = `http://127.0.0.1:8000/getTable/\${tableSpecifier}`

The VS Code interface includes tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and JUPYTER. The TERMINAL tab shows a log of HTTP requests and responses, including errors for favicon.ico and broken pipe issues.

# 1st October 2022 (01/10/2022)

The screenshot shows a developer's workspace with two main windows: a browser and a code editor.

**Browser Window:** Displays the 'Planify' application at <http://127.0.0.1:8000>. The interface includes a navigation bar with tabs for 'Planify', 'Get Nodes - D...', 'Get Subjects - I...', 'python - D...', 'Serializers - I...', 'Task List - Dj...', 'Select week to ...', and a '+' button. Below the navigation is a search bar with the URL. The main content area shows a table with rows for 'Exam', 'Homework', 'Sleep', 'Fail', 'Revise', and 'Business'. Each row has three columns: 'true' (highlighted in blue), 'false', and 'false'. A button labeled 'Add Week' is visible above the table. At the bottom of the page is a footer with links for 'Inspector', 'Console', 'Debugger', 'Network', 'Style Editor', 'Performance', 'Memory', 'Storage', and a status bar showing '127.0.0.1:8000:96:21'.

**Code Editor:** Shows the source code for 'home.html' in Visual Studio Code. The code contains JavaScript logic for handling button events and making API requests to 'http://127.0.0.1:8000'. It includes functions for updating nodes, adding weeks, and adding subjects. The code uses variables like 'nodeButtons', 'addWeekBtn', and 'addSubjectBtn' along with 'fetch' and 'JSON.stringify' methods.

**Bottom Status Bar:** Provides information about the current session: 'Ln 173, Col 58, Spaces: 4, UTRF-8, CRLF, HTML, Go Live'. The date and time '13:28 01/10/2022' are also displayed.

# 2<sup>nd</sup> October 2022 (02/10/2022)

The screenshot shows a web browser window with multiple tabs open. The active tab is 'Planify' at [127.0.0.1:8000](http://127.0.0.1:8000). The page displays a table with rows for Exam, Homework, Sleep, Fail, Revise, and Business, each with two columns labeled 'Week 1' and 'Week 2'. A modal window titled 'Add Subject' is open in the center, containing a text input field labeled 'Subject name:' and two buttons: 'Close' and 'Save changes'. To the right of the browser window is an instance of Visual Studio Code. The code editor has 'home.html' selected and is displaying the following JavaScript code:

```
base > templates > base > home.html > script > displayTables
145     complete_table[complete_table.length - 1][witer] = node_data[nIter]
146
147 }
148
149
150
151
152 instruct HTML representation of above table
153 item = '<table>'
154 console.log(complete_table)
155
156 ble_and_weeks
157 += `<tr>
158 += `<th>${(complete_table[0][0])}</th>
159 var i = 1; i < complete_table[0].length; i++
160
161 item += `<th>Week ${complete_table[0][1].number}</th>
162
163 += `<th><button id="add-week-btn">Add Week</button></th>
164 += `</tr>
165
166 objects_and_nodes
167 var sIter = 1; sIter < complete_table.length; sIter++
168
169 item += `<tr>
170 item += `<td>${(complete_table[sIter][0])}</td>
171 / Nodes
172 or (var nIter = 1; nIter < complete_table[0].length; nIter++)
173
174 item += `<td><button class="node">${(complete_table[sIter][nIter]).completed}</button></td>
175
176 item += `</tr>
177
178 += `<tr><td><button id="add-subject-btn" class="btn btn-primary" data-toggle="modal" data-target="#exampleModal">A
179
180 += `</tr>
181
182 _list.innerHTML += item
183
184 // Adding listeners to nodes - TEST
185 nodeButtons = document.getElementsByClassName('node')
186 nodeIndex = 0
187 var sIter = 1; sIter < complete_table.length; sIter++
188
189 or (var nIter = 1; nIter < complete_table[sIter].length; nIter++)
190`
```

The bottom status bar of the VS Code window shows: Line 179, Column 130, Spaces: 4, CRLF, HTML, Go Live, and the date/time 02/10/2022 10:47.

**2<sup>nd</sup> October 2022 (02/10/2022)**

# 2nd October 2022 (02/10/2022)

The screenshot shows a web application titled "Planify" running on a local server at 127.0.0.1:8000. The application interface includes a navigation bar with tabs for "Planify", "TypeError is undefined", "How to add a value to", and "Set value of textarea in". Below the tabs, there's a header with a search icon and other UI elements.

The main content area displays a weekly schedule grid. The columns are labeled "Week 1", "Week 2", "Week 3", "Week 4", "Week 5", "Week 6", and "Add Week". The rows represent different subjects: Exam, Homework, Sleep, Fail, Revise, Business, Business, nonono, and test123. Each cell in the grid contains a boolean value (true or false). A blue button labeled "Add Subject" is located at the bottom left of the grid.

On the right side of the screen, the Visual Studio Code editor is open, showing the file "home.html". The code contains JavaScript functions for updating subjects and renaming them. It uses fetch() to interact with a backend API at "http://127.0.0.1:8000". The "renameSubject" function appends the new name to the subject name input and triggers an event listener on the "rename-subject" element. The "getTable" function retrieves a table from the API based on a specifier.

At the bottom of the screen, the browser's developer tools are visible, showing the network tab with a request to "GET http://127.0.0.1:8000/favicon.ico" and the console tab with log entries indicating successful API calls and a 404 error for the favicon.

**5<sup>th</sup> October 2022 (05/10/2022)**

The screenshot shows a developer's environment with multiple windows open. On the left, a web browser window displays a project management application titled 'Planify' with a 7x8 grid of status indicators for various tasks over a week. The tasks listed on the left are Exam, Homework, Sleep, Fail, Revise, Business, Lecture, nonono, ha, and meow. The status grid contains mostly 'false' values. Below the grid is a button labeled 'Add Subject'. At the top of the browser window, several tabs are visible: '00:01:01', 'Planify', 'Get Subjects - Django', 'Get Weeks - Django', 'Get Nodes - Django', and '+'. The address bar shows '127.0.0.1:8000'. On the right, a Visual Studio Code editor window is open, showing the 'home.html' file. The code includes JavaScript logic for handling form submissions, managing cookies, and performing asynchronous requests to 'getSubjects', 'getWeeks', 'getNodes', and 'getData' endpoints. The code also includes a setTimeout function to delay the execution of 'displayTables'. The bottom of the screen shows the terminal output with numerous log entries from October 5, 2022, at 09:40:52, detailing the execution of these requests. The system tray at the bottom left shows icons for File Explorer, Task View, Microsoft Edge, and other system utilities.

**5<sup>th</sup> October 2022 (05/10/2022)**

The screenshot displays a dual-monitor setup. The left monitor shows a web browser window titled 'Planify' at the URL '127.0.0.1:8000'. The page contains a header 'Planify' and a table with columns 'Project 1' and 'Week 1' through 'Week 8'. Rows include 'Exam', 'Homework', 'Sleep', 'Fail', 'Revise', 'Business', 'Lecture', 'nonono', 'ha', 'meow', and 'hahaha', each with a series of boolean values (true/false) across the weeks. A blue button 'Add Week' is located at the top right of the table, and a blue button 'Add Subject' is at the bottom left. The right monitor shows the Visual Studio Code interface with the file 'home.html' open. The code is a JavaScript file containing functions for handling table data, subjects, weeks, and nodes. It includes comments for an IDEA and a setTimeout function for performing actions after data retrieval. The code editor shows syntax highlighting for JavaScript. Below the code editor are tabs for 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL', and 'JUPYTER'. The terminal tab shows a command-line session with an error message about an index out of range. The status bar at the bottom indicates the file is 'home.html - planify - Visual Studio Code'.

# 5<sup>th</sup> October 2022 (05/10/2022)

The screenshot shows a dual-pane interface. On the left is a web browser window titled "Planify" displaying a Django application at [127.0.0.1:8000](http://127.0.0.1:8000). The page has a header "Planify" and a navigation bar with tabs: "Algorithms", "Week 1", "Week 2", and "Add Week". Under "Algorithms", there are two buttons: "Big O" and "true". Below the tabs is a "Create new table" button. On the right is a "Visual Studio Code" window showing the file "home.html". The code is a JavaScript file containing logic for handling table creation and deletion. It includes functions for creating tables via POST requests to `'http://127.0.0.1:8000/createTable'` and deleting nodes via POST requests to `'/updateNode/217'`. The code uses jQuery-style syntax and includes comments explaining the functionality. The bottom of the screen shows the terminal output with several log entries from the application's server.

```
base > templates > base > home.html > html > script > displayTables() > then() callback
    'Content-type': 'application/json',
    'X-CSRFToken': csrfToken,
),
body:JSON.stringify({ 'name':subjectName.value, 'table':currentTable}),
)
).then(function(response)
{
    //displayTables()
    window.location.reload()
})
)
)
subjectName.value = null

// Create table button Listener
var createTableBtn = document.getElementById('create-table')
var tableName = document.getElementById('table-name-input-field')
createTableBtn.addEventListener('click', function(){
    var url = 'http://127.0.0.1:8000/createTable'
    fetch(url, {
        method: 'POST',
        headers: {
            'Content-type': 'application/json',
            'X-CSRFToken': csrfToken,
        },
        body:JSON.stringify({ 'name':tableName.value})
    )
    .then(function(response)
    {
        //displayTables()
        window.location.reload()
    })
)
)
tableName.value = null
}

// NOT SUPPOSED TO DELETE NODES DIRECTLY
/*function deleteNode(node)
{
    console.log("Node pressed: ", node)
}*/



// delete functionality:
// - whilst iterating in above code, match the representation to actual object
// - grab the class names attached and add event listeners
// For this to work you need to always add the class to the node

```

Terminal Output:

```
[05/Oct/2022 11:27:12] "GET /getSubjects/5/ HTTP/1.1" 200 36
[05/Oct/2022 11:27:12] "GET /getTable/5/ HTTP/1.1" 200 30
[05/Oct/2022 11:27:12] "GET /getWeeks/5/ HTTP/1.1" 200 63
[05/Oct/2022 11:27:12] "GET /getNodes/5/ HTTP/1.1" 200 104
[05/Oct/2022 11:27:12] "POST /updateNode/217/ HTTP/1.1" 200 50
[05/Oct/2022 11:27:12] "GET / HTTP/1.1" 200 25955
[05/Oct/2022 11:27:12] "GET /getTable/5/ HTTP/1.1" 200 30
[05/Oct/2022 11:27:12] "GET /getWeeks/5/ HTTP/1.1" 200 63
[05/Oct/2022 11:27:12] "GET /getSubjects/5/ HTTP/1.1" 200 36
[05/Oct/2022 11:27:12] "GET /getNodes/5/ HTTP/1.1" 200 103
```

# 8<sup>th</sup> October 2022 (08/10/2022)

The screenshot shows a development environment with two main windows. On the left is a browser window displaying a web application named "Planify". The application has a navigation bar with tabs: "Algorithms", "Week 1", "Week 2", and "Add Week". Under "Algorithms", there are four categories: "Study", "Homework", and "Add Subject". "Study" and "Homework" each have a table with five columns labeled "Week 1" through "Week 5". The first column for both categories contains the value "false". Under "Add Subject", there are four subjects: "AI", "Robot", "Smart", and "Learn", each with a table of the same structure. Below these sections is a "Create new table" button. The browser's developer tools are open, showing the Network tab with several requests made to "http://127.0.0.1:8000".

The right window is a code editor for Visual Studio Code, showing three files: "views.py", "home.html", and "urls.py". The "home.html" file contains script tags that interact with the tables in the application. The "urls.py" file defines URLs for "getTables", "getTable/1", "getSubjects/1", "getWeeks/1", "getNode/1", "getTable/2", "getSubjects/2", and "getWeeks/2". The "views.py" file contains logic for generating tables based on subject names and handling interactions like clicks.

```
File Edit Selection View Go Run Terminal Help home.html - files - Visual Studio Code
views.py
base > templates > base > home.html > html > script > displayTables > <function> > <function>
333 let subjectButtons = document.getElementsByClassName('subject->tIter')
334 let subjectIndex = 0
335 for (let sIter = 1; sIter < completeTables[tIter].length; sIter++)
336 {
337     let subjectName = completeTables[tIter][sIter][0]
338     subjectButtons[subjectIndex].addEventListener('click', (function(subject){
339         return function(){
340             subjectInteractions(subject)
341         }
342     })(subjectName))
343     subjectIndex += 1
344 }
345
346 // Adding Listeners to weeks
347 let weekButtons = document.getElementsByClassName('week->tIter')
348 let weekIndex = 0
349 for (let wIter = 1; wIter < completeTables[tIter][0].length; wIter++)
350 {
351     let week = completeTables[tIter][0][wIter]
352     weekButtons[wIter].addEventListener('click', function(week){
353         return function(){
354             weekInteractions(week)
355         }
356     })(week)
357     weekIndex += 1
358 }
359
360
361 function tableInteractions(table)
362 {
363     console.log(table.name)
364 }
365
366 function subjectInteractions(subject)
367 {
368     console.log(subject.name)
369 }
370
371 function weekInteractions(week)
372 {
373     console.log(week.number)
374 }
375
376 </script>
377 </html>
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

[08/Oct/2022 14:59:38] "GET / HTTP/1.1" 200 16965  
[08/Oct/2022 14:59:39] "GET /api/getTables/ HTTP/1.1" 200 51  
[08/Oct/2022 14:59:39] "GET /api/getTable/1/ HTTP/1.1" 200 30  
[08/Oct/2022 14:59:39] "GET /api/getSubjects/1/ HTTP/1.1" 200 72  
[08/Oct/2022 14:59:39] "GET /api/getWeeks/1/ HTTP/1.1" 200 61  
[08/Oct/2022 14:59:39] "GET /api/getNodes/1/ HTTP/1.1" 200 193  
[08/Oct/2022 14:59:39] "GET /api/getTable/2/ HTTP/1.1" 200 22  
[08/Oct/2022 14:59:39] "GET /api/getSubjects/2/ HTTP/1.1" 200 103  
[08/Oct/2022 14:59:39] "GET /api/getWeeks/2/ HTTP/1.1" 200 151  
[08/Oct/2022 14:59:39] "GET /api/getNodes/2/ HTTP/1.1" 200 731

Ln 354, Col 51 Spaces: 4 UTRF HTML Go Live

14:59 08/10/2022

**25<sup>th</sup> May 2023 (25/05/2023)**

The screenshot shows a developer's environment with multiple windows open:

- Browser:** A Firefox window titled "Planyfy - Test Page" showing a simple table with three rows. The first row has "TableID=275 Number=0" and buttons "Modify table 275" and "Delete table 275". The second row has "TableID=276 Number=1" and buttons "Modify table 276" and "Delete table 276". The third row has "TableID=277 Number=8" and buttons "Modify table 277" and "Delete table 277".
- Code Editor:** An integrated code editor in Visual Studio Code showing the file `home.html`. The code includes JavaScript functions for handling table data and interacting with an API.
- Terminal:** A terminal window at the bottom showing a log of HTTP requests from May 25, 2023, at 08:55:29, all resulting in status code 200.

**25<sup>th</sup> May 2023 (25/05/2023)**

The screenshot shows a web-based study tracker named "Planify". The interface includes a header with the logo, title, and navigation links for "Sign Up" and "Login". Below the header is a toolbar with buttons for "Algorithms", "Week 1", "Week 2", "Week 3", "Week 4", and "Add Week". A main table displays study progress for subjects like "Study", "Homework", "Big O", "Hi", and "Meow" across weeks 1 through 4. Each subject has a row of four cells, each containing either "true" or "false". Below this table is another section with a "Add Subject" button, a "Week 1" to "Week 18" toolbar, and rows for "AI", "Huh", and "Smart". Each of these subjects has a row of 18 cells corresponding to the weeks. Further down are sections for "Enigma" (with "Week 1" and "Add Week" buttons) and "Code" (with two cells). There is also a "Create new table" button. The bottom of the screen shows the Windows taskbar with various pinned icons and the system clock indicating 09:38 on 25/05/2023.

27<sup>th</sup> May 2023 (27/05/2023)

The screenshot shows a web browser window displaying the Planify Study Tracker application at [127.0.0.1:8000](http://127.0.0.1:8000). The application interface includes a navigation bar with 'Sign Up' and 'Login' buttons, and a sidebar with buttons for 'Algorithms', 'Study', 'Assignment', 'Big O', 'Java', 'Programming', 'Developing', 'Add Subject', 'Training', 'Byeee', 'Helloooo', 'Add Subject', and 'Create new table'. The main content area contains two tables. The first table has columns for 'Week 1' through 'Week 5' and an 'Add Week' button. The second table has columns for 'Week 1' through 'Week 18' and an 'Add Week' button. Both tables use a color-coded system where green indicates true values and red indicates false values. A modal window titled 'Subject Create' is visible in the background, containing fields for 'Subject Name' and 'Subject Description'. The browser's developer tools are open, showing the HTML structure and CSS styles for the application.

	Week 1	Week 2	Week 3	Week 4	Week 5	Add Week
Algorithms	true	true	true	false	false	
Study	true	true	false	false	false	
Assignment	true	true	false	false	false	
Big O	true	true	false	false	true	
Java	true	true	true	true	false	
Programming	false	false	false	true	true	
Developing	false	false	true	false	true	
Add Subject						

	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12	Week 13	Week 14	Week 15	Week 16	Week 17	Add Week
AI																		
Huh	true	true	false	false	false	true	true	false	true	false	true	false	true	false	false	false	false	
Smart	false	true	true	false	false	true	false	true	true	false	true	false	true	false	false	false	false	
Revise	false	false	false	true	false	true	false	true	false	false	true	false	false	false	false	false	false	
Add Subject																		
Training																		
Byeee	false	true	false	true	true	true	false	false	false	false	false	false	false	false	false	false	false	
Helloooo	false	true	true	true	false	true	false	false	false	false	false	false	false	false	false	false	false	
Add Subject																		
Create new table																		

Developer Tools (Inspector) showing the HTML structure and CSS styles:

```
<!DOCTYPE html>
<html> [event]
  <head> [event]
    <meta charset="UTF-8">
    <title>Planify - Study Tracker</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-alpha1/dist/css/bootstrap.min.css" integrity="sha384-oesi62ohfzrys4LxRf6303CxDlDipYBnV19y9/T1w5xlKEHpyvvDShgf/" crossorigin="anonymous">
    <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.5.4/dist/umd/popper.min.js" integrity="sha384-Q6E9RHBIYzZFJoft+2mJbHAEld1vriOy5n3zv9zTtmI3UksdQRVvoxMfooAo" crossorigin="anonymous"></script>
    <script src="https://stackpath.bootstrapcdn.com/bootstrap/5.0.0-alpha1/js/bootstrap.min.js" integrity="sha384-oesi62ohfzrys4LxRf6303CxDlDipYBnV19y9/T1w5xlKEHpyvvDShgf/" crossorigin="anonymous"></script>
  </head>
  <body>
    <nav class="navbar navbar-expand-sm">[event]
      <!--Anchor for all tables-->
      <div id="tables-root">[event]
        <!--Modal For Subject Add-->
        <div id="subjectCreateModal" class="modal fade" tabindex="-1" role="dialog" aria-labelledby="subjectCreateModalLabel" aria-hidden="true">[event]
          <!--Modal For Subject Interactions-->
          <div id="subjectInteractions" class="modal fade" tabindex="-1" role="dialog" aria-labelledby="subjectInteractionsLabel" aria-hidden="true">[event]
            <!--Modal For Table Interactions-->
            <div id="tableInteractions" class="modal fade" tabindex="-1" role="dialog" aria-labelledby="tableInteractionsLabel" aria-hidden="true">[event]
              <!--Modal For Week Interactions-->
              <div id="weekInteractions" class="modal fade" tabindex="-1" role="dialog" aria-labelledby="weekInteractionsLabel" aria-hidden="true">[event]
                <!--Modal For Table Create-->
                <div id="tableCreateInteractions" class="modal fade" tabindex="-1" role="dialog" aria-labelledby="tableCreateInteractionsLabel" aria-hidden="true">[event]
                  <!--Button to display table creation modal-->
                  <button id="create-table-btn" class="btn btn-primary" data-toggle="modal" data-target="#tableCreateInteractions">Create new table</button>
                  <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.5.4/dist/umd/popper.min.js" integrity="sha384-Q6E9RHBIYzZFJoft+2mJbHAEld1vriOy5n3zv9zTtmI3UksdQRVvoxMfooAo" crossorigin="anonymous"></script>
                  <script src="https://stackpath.bootstrapcdn.com/bootstrap/5.0.0-alpha1/js/bootstrap.min.js" integrity="sha384-oesi62ohfzrys4LxRf6303CxDlDipYBnV19y9/T1w5xlKEHpyvvDShgf/" crossorigin="anonymous"></script>
                </div>
              </div>
            </div>
          </div>
        </div>
      </div>
      <!--Table Interactions-->
      <div id="tableInteractions" class="modal fade" tabindex="-1" role="dialog" aria-labelledby="tableInteractionsLabel" aria-hidden="true">[event]
        <!--Week Interactions-->
        <div id="weekInteractions" class="modal fade" tabindex="-1" role="dialog" aria-labelledby="weekInteractionsLabel" aria-hidden="true">[event]
          <!--Table Create-->
          <div id="tableCreateInteractions" class="modal fade" tabindex="-1" role="dialog" aria-labelledby="tableCreateInteractionsLabel" aria-hidden="true">[event]
            <!--Button to display table creation modal-->
            <button id="create-table-btn" class="btn btn-primary" data-toggle="modal" data-target="#tableCreateInteractions">Create new table</button>
            <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.5.4/dist/umd/popper.min.js" integrity="sha384-Q6E9RHBIYzZFJoft+2mJbHAEld1vriOy5n3zv9zTtmI3UksdQRVvoxMfooAo" crossorigin="anonymous"></script>
            <script src="https://stackpath.bootstrapcdn.com/bootstrap/5.0.0-alpha1/js/bootstrap.min.js" integrity="sha384-oesi62ohfzrys4LxRf6303CxDlDipYBnV19y9/T1w5xlKEHpyvvDShgf/" crossorigin="anonymous"></script>
          </div>
        </div>
      </div>
    </nav>
    <div id="tables-root">[event]
      <!--Modal For Subject Add-->
      <div id="subjectCreateModal" class="modal fade" tabindex="-1" role="dialog" aria-labelledby="subjectCreateModalLabel" aria-hidden="true">[event]
        <!--Modal For Subject Interactions-->
        <div id="subjectInteractions" class="modal fade" tabindex="-1" role="dialog" aria-labelledby="subjectInteractionsLabel" aria-hidden="true">[event]
          <!--Modal For Table Interactions-->
          <div id="tableInteractions" class="modal fade" tabindex="-1" role="dialog" aria-labelledby="tableInteractionsLabel" aria-hidden="true">[event]
            <!--Modal For Week Interactions-->
            <div id="weekInteractions" class="modal fade" tabindex="-1" role="dialog" aria-labelledby="weekInteractionsLabel" aria-hidden="true">[event]
              <!--Modal For Table Create-->
              <div id="tableCreateInteractions" class="modal fade" tabindex="-1" role="dialog" aria-labelledby="tableCreateInteractionsLabel" aria-hidden="true">[event]
                <!--Button to display table creation modal-->
                <button id="create-table-btn" class="btn btn-primary" data-toggle="modal" data-target="#tableCreateInteractions">Create new table</button>
                <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.5.4/dist/umd/popper.min.js" integrity="sha384-Q6E9RHBIYzZFJoft+2mJbHAEld1vriOy5n3zv9zTtmI3UksdQRVvoxMfooAo" crossorigin="anonymous"></script>
                <script src="https://stackpath.bootstrapcdn.com/bootstrap/5.0.0-alpha1/js/bootstrap.min.js" integrity="sha384-oesi62ohfzrys4LxRf6303CxDlDipYBnV19y9/T1w5xlKEHpyvvDShgf/" crossorigin="anonymous"></script>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </body>

```

HTML > body

Changes

No changes found.

Changes to CSS in Inspector will appear here.

27<sup>th</sup> May 2023 (27/05/2023)

The screenshot shows a web-based application titled "Planify Study Tracker". The interface features a navigation bar with "Sign Up" and "Login" buttons. On the left, there's a sidebar with buttons for "Algorithms", "Study", "Assignment", "Big O", "Java", "Programming", "Developing", and "Add Subject". The main area displays a grid of colored circles representing a study plan across five weeks. Below this, another grid shows a 17-week plan for subjects like "AI", "Huh", "Smart", "Revise", and "Training". At the bottom, there's a footer with various icons and a timestamp "21:06 27/05/2023".

Planify Study Tracker

Sign Up Login

Algorithms Week 1 Week 2 Week 3 Week 4 Week 5 Add Week

Study Assignment Big O Java Programming Developing Add Subject

AI Week 1 Week 2 Week 3 Week 4 Week 5 Week 6 Week 7 Week 8 Week 9 Week 10 Week 11 Week 12 Week 13 Week 14 Week 15 Week 16 Week 17 Week 18 Add Week

Huh Smart Revise Add Subject

Training Week 1 Week 2 Week 4 Week 9 Week 10 Week 11 Week 12 Week 15 Week 16 Week 17 Add Week

Byeee Helloooo

Inspector Console Debugger Network

Search HTML

```
<!DOCTYPE html>
<html> event scroll
  <head> </head>
  <body> overflow
    > <nav class="navbar navbar-expand-sm" style="background-color: #333; color: white; padding: 5px; border-bottom: 1px solid black;"> </nav> flex
      <!-- Anchor for all tables -->
      > <div id="tables-root" style="flex-grow: 1; margin: 10px; border: 1px solid #ccc; padding: 10px; border-radius: 5px;"> </div>
      <!-- Model For Subject Add -->
      > <div id="subjectCreateModal" class="modal fade" tabindex="-1" role="dialog" aria-labelledby="subjectCreateModalLabel" aria-hidden="true"> </div>
      <!-- Model For Subject Interactions -->
      > <div id="subjectInteractions" class="modal fade" tabindex="-1" role="dialog" aria-labelledby="subjectInteractionsLabel" aria-hidden="true"> </div>
      <!-- Model For Table Interactions -->
      > <div id="tableInteractions" class="modal fade" tabindex="-1" role="dialog" aria-labelledby="tableInteractionsLabel" aria-hidden="true"> </div>
      <!-- Model For Week Interactions -->
      > <div id="weekInteractions" class="modal fade" tabindex="-1" role="dialog" aria-labelledby="weekInteractionsLabel" aria-hidden="true"> </div>
      <!-- Model For Table Create -->
      > <div id="tableCreateInteractions" class="modal fade" tabindex="-1" role="dialog" aria-labelledby="tableCreateInteractionsLabel" aria-hidden="true"> </div>
      <!-- Button to display table creation modal -->
      > <button id="create-table-btn" class="btn btn-primary" data-toggle="modal" data-target="#tableCreateInteractions">Create new table</button> overflow
      <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.16.0/dist/umd/popper.min.js" integrity="sha384-Q6E9RHViZMvKbV95PjN5kNq1juTdG8RQy+u+L7xWzJfKXlqJgkH+oqBQf+Xg==" crossorigin="anonymous"></script>
      <script src="https://stackpath.bootstrapcdn.com/bootstrap/5.0.0-alpha1/js/bootstrap.min.js" integrity="sha384-oesi2h0LfrzS4LxF630JCXxDipipiWBnVl9Y9/Tl1w5x1KEHpvyyDShgF/" crossorigin="anonymous"></script>
      <script> </script>
    </body>
```

html > body

Filter Style Show .ds + Layout Computed Changes

No changes found.

Changes to CSS in Inspector will appear here.

element ::{ inline } body ::{ \_reboot.scss:40 margin: 0; font-family: var(--bs-font-sans-serif); font-size: 1rem; font-weight: 400; line-height: 1.5; color: #212529; background-color: #fff; -webkit-text-size-adjust: 100%; -webkit-tap-highlight-color: transparent; }

\*, ::after, ::before { bootstrap.css:36 box-sizing: border-box; }

21:06 27/05/2023

27<sup>th</sup> May 2023 (27/05/2023)

The screenshot shows a web-based study tracker application titled "Planity Study Tracker". The interface features a sidebar on the left with subject categories: Algorithms, Study, Assignment, Big O, Java, Programming, Developing, and Add Subject. Below this is a "Weeks" section with buttons for Week 1 through Week 17, and an "Add Week" button. The main area displays three separate tables, each consisting of a header row with week numbers and a body row with circular markers representing study progress. The first table (Algorithms) has 5 rows of markers. The second table (Study) has 7 rows of markers. The third table (Assignment) has 7 rows of markers. The fourth table (Big O) has 7 rows of markers. The fifth table (Java) has 7 rows of markers. The sixth table (Programming) has 7 rows of markers. The seventh table (Developing) has 7 rows of markers. The eighth table (Training) has 7 rows of markers. The ninth table (Byeee) has 7 rows of markers. The tenth table (Helloooo) has 7 rows of markers. The application is running on a local server at 127.0.0.1:8000.

Planity Study Tracker

Sign Up Login

Algorithms Week 1 Week 2 Week 3 Week 4 Week 5 Add Week

Study

Assignment

Big O

Java

Programming

Developing

Add Subject

Week 1 Week 2 Week 3 Week 4 Week 5 Week 6 Week 7 Week 8 Week 9 Week 10 Week 11 Week 12 Week 13 Week 14 Week 15 Week 16 Week

Training Week 1 Week 2 Week 4 Week 9 Week 10 Week 11 Week 12 Week 15 Week 16 Week 17 Add Week

Byeee

Helloooo

Add Subject

Create new table

Inspector Console Debugger Network

Search HTML

```
<!DOCTYPE html>
<html> <event scroll>
<head></head>
<body>
```

Layout Computed Changes

No changes found.

Changes to CSS in Inspector will appear here.

21:28 27/05/2023

27<sup>th</sup> May 2023 (27/05/2023)

The screenshot shows a web-based study tracker application named "Planify". The interface is organized into several sections:

- Top Bar:** Includes tabs for "Planify - Study Tracker", "Modal Object Creation.", "W3Schools Tryit Editor", "Color Hex Color Codes", "Buttons - Bootstrap v5.0", and "CSS font-weight property".
- Header:** Features the "Planify" logo and "Study Tracker" text, along with "Sign Up" and "Login" buttons.
- Algorithm Section:** Shows a grid of colored circles representing study progress for subjects like "Study", "Assignment", "Big O", "Java", "Programming", and "Developing" over "Week 1" through "Week 5". A blue button "+ Add Week" is visible.
- AI Section:** Shows a grid of colored circles for subjects "Huh", "Smart", and "Revise" over "Week 1" through "Week 15". A blue button "+ Add Subject" is visible.
- Training Section:** Shows a grid of colored circles for subjects "Byeee" and "Helloooo" over "Week 1" through "Week 17". A blue button "+ Add Week" is visible.
- General Subjects:** A section labeled "+ Add Subject" contains subjects like "Create new table".
- Right Sidebar:** Contains the browser's developer tools (Inspector). The "Elements" tab is selected, showing the DOM structure: <!DOCTYPE html> <html> <head> </head> <body> </body> </html>. The "Changes" tab is also present, stating "No changes found." and "Changes to CSS in Inspector will appear here."

The bottom of the screen shows the Windows taskbar with various pinned icons and the system tray indicating the date and time as 27/05/2023 at 22:12.

27<sup>th</sup> May 2023 (27/05/2023)

Planify - Study Tracker    Modal Object Creation.    W3Schools Tryit Editor    Color Hex Color Codes    Buttons - Bootstrap v5.0    CSS font-weight property    CSS Margin vs. Padding: What's the Difference?

127.0.0.1:8000

# Planify

Study Tracker

Sign Up   Login

Algorithms

	Week 1	Week 2	Week 3	Week 4	Week 5	+ Add Week
Study	Green	Green	Green	Red	Red	
Assignment	Green	Green	Red	Red	Red	
Big O	Green	Green	Red	Red	Green	
Java	Green	Green	Green	Green	Red	
Programming	Red	Red	Red	Green	Green	
Developing	Red	Red	Green	Red	Green	

+ Add Subject

AI

	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12	Week 13	Week 14	Week 15	Week 16	Week 17	Week 18	Week 19	Week 20	Week 21
Huh	Green	Green	Red	Red	Red	Green	Red	Green	Red	Red	Green	Red	Red	Green	Red	Red	Red	Red	Red	Red	
Smart	Red	Red	Red	Red	Red	Green	Red	Green	Red	Red	Green	Red	Red	Green	Red	Red	Red	Red	Red	Red	
Revise	Red	Red	Red	Red	Red	Green	Red	Green	Red	Red	Green	Red	Red	Green	Red	Red	Red	Red	Red	Red	

+ Add Subject

Training

	Week 1	Week 2	Week 4	Week 9	Week 10	Week 11	Week 12	Week 15	Week 16	Week 17	+ Add Week
Byeee	Red	Green	Red	Green	Red	Green	Red	Red	Red	Red	
Hellooooo	Red	Green	Red	Green	Red	Green	Red	Red	Red	Red	

+ Add Subject

Windows Taskbar icons: File Explorer, File Manager, Microsoft Edge, File Explorer, Spotify, File Explorer, Planify - Study Tracker, Task View, Volume, Network, Battery, 22:25, 27/05/2023, 24 notifications.

**28<sup>th</sup> May 2023 (28/05/2023)**

**28<sup>th</sup> May 2023 (28/05/2023)**

Planify - Study Tracker × html - Placing border inside of × Modal Object Creation. × CSS Text Decoration × W3Schools Tryit Editor × +

127.0.0.1:8000 ☆

Sign Up Login

# Planify

Study Tracker

Algorithms Week 1 Week 2 Week 3 Week 4 Week 5 + Add Week

Study	Green	Green	Green	Red	Red
Assignment	Red	Green	Red	Green	Red
Big O	Green	Red	Red	Red	Green
Java	Green	Green	Green	Green	Red
Programming	Red	Red	Red	Green	Green
Developing	Green	Red	Green	Red	Green

+ Add Subject

AI Week 1 Week 2 Week 3 Week 4 Week 5 Week 6 Week 7 Week 8 Week 9 Week 10 Week 11 Week 12 Week 13 Week 14 Week 15 Week 16 Week 17 Week 18 Week 19 Week 20 Week 21

Huh	Green	Green	Red	Red	Red	Green	Green	Red	Green	Red	Red	Green	Red	Red	Red	Red	Green	Red	Red	Red
Smart	Red	Green	Green	Red	Green	Red	Green	Red	Green	Red	Red	Green	Red	Red	Red	Red	Green	Red	Red	Red
Revise	Red	Red	Red	Red	Red	Green	Red	Red	Red	Red	Red	Green	Red	Red	Red	Red	Red	Green	Red	Red

+ Add Subject

---

Training Week 1 Week 2 Week 4 Week 9 Week 10 Week 11 Week 12 Week 15 Week 16 Week 17 + Add Week

Byeee	Red	Green	Red	Green	Red	Green	Red	Red	Red	Red
Hellooooo	Red	Green	Red	Green	Red	Green	Red	Red	Red	Red

+ Add Subject

09:50 24  
28/05/2023

**28<sup>th</sup> May 2023 (28/05/2023)**

Planify - Study Tracker X Modal Object Creation. X W3Schools Tryit Editor X Daniooo X +

127.0.0.1:8000 ☆

My Account 

# Planify Study Tracker

"Compare yourself to who you were yesterday, not to who someone else is today" - Jordan Peterson >

Algorithms Week 1 Week 2 Week 3 Week 4 Week 5 Week 6 Week 7 + Add Week 99%

	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7
Study	Green	Green	Green	Red	Red	Green	Red
Assignment	Red	Green	Red	Green	Red	Red	Red
Big O	Green	Red	Red	Red	Green	Green	Green
Java	Green	Green	Green	Green	Red	Red	Red
Programming	Red	Red	Red	Green	Green	Green	Red
Developing	Green	Red	Green	Red	Green	Red	Green

+ Add Subject

AI Week 1 Week 2 Week 3 Week 4 Week 5 Week 6 Week 7 Week 8 Week 9 Week 10 Week 11 Week 12 Week 13 Week 14 Week 15 Week 16 Week 17 + Add Week 99%

	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12	Week 13	Week 14	Week 15	Week 16	Week 17
Huh	Green	Green	Red	Red	Red	Green	Green	Red	Green	Red	Green	Red	Red	Green	Red	Red	Red
Smart	Red	Green	Green	Red	Green	Green	Red	Green	Green	Red	Red	Red	Green	Red	Green	Red	Red
Revise	Red	Red	Red	Red	Red	Green	Red	Green	Red	Red	Green	Red	Red	Green	Red	Red	Red

+ Add Subject

Training Week 1 Week 2 Week 4 Week 9 Week 10 Week 11 Week 12 Week 15 Week 16 Week 17 + Add Week 99%

	Week 1	Week 2	Week 4	Week 9	Week 10	Week 11	Week 12	Week 15	Week 16	Week 17
Byeee	Red	Green	Red	Green	Green	Green	Red	Red	Red	Red
Hellooooo	Red	Green	Green	Green	Red	Green	Red	Green	Red	Red

+ Add Subject

18:31 26 28/05/2023

**28<sup>th</sup> May 2023 (28/05/2023)**

**28<sup>th</sup> May 2023 (28/05/2023)**

Planify - Study Tracker × Modal Object Creation. × W3Schools Tryit Editor × Daniooo × +

127.0.0.1:8000

My Account 

# Planify

Study Tracker

"Don't stop when you're tired, stop when you're done." - Wesley Snipes

Algorithms

	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	+ Add Week
Study	Green	Green	Green	Red	Red	Green	Red	
Assignment	Red	Green	Red	Green	Red	Red	Red	
Big O	Green	Red	Red	Red	Green	Green	Green	
Java	Green	Green	Green	Green	Red	Red	Red	
Programming	Red	Red	Red	Green	Green	Red	Red	
Developing	Green	Red	Green	Red	Green	Red	Green	

99%

+ Add Subject

AI

	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12	Week 13	Week 14	Week 15	Week 16	Week 17	+ Add Week
Huh	Green	Green	Red	Red	Red	Green	Red	Red	Green	Red								
Smart	Red	Green	Green	Green	Red	Green	Green	Green	Green	Red								
Revise	Red	Green	Red	Red	Red	Green	Red	Red	Red	Red	Red	Red	Red	Red	Red	Red	Red	

99%

+ Add Subject

---

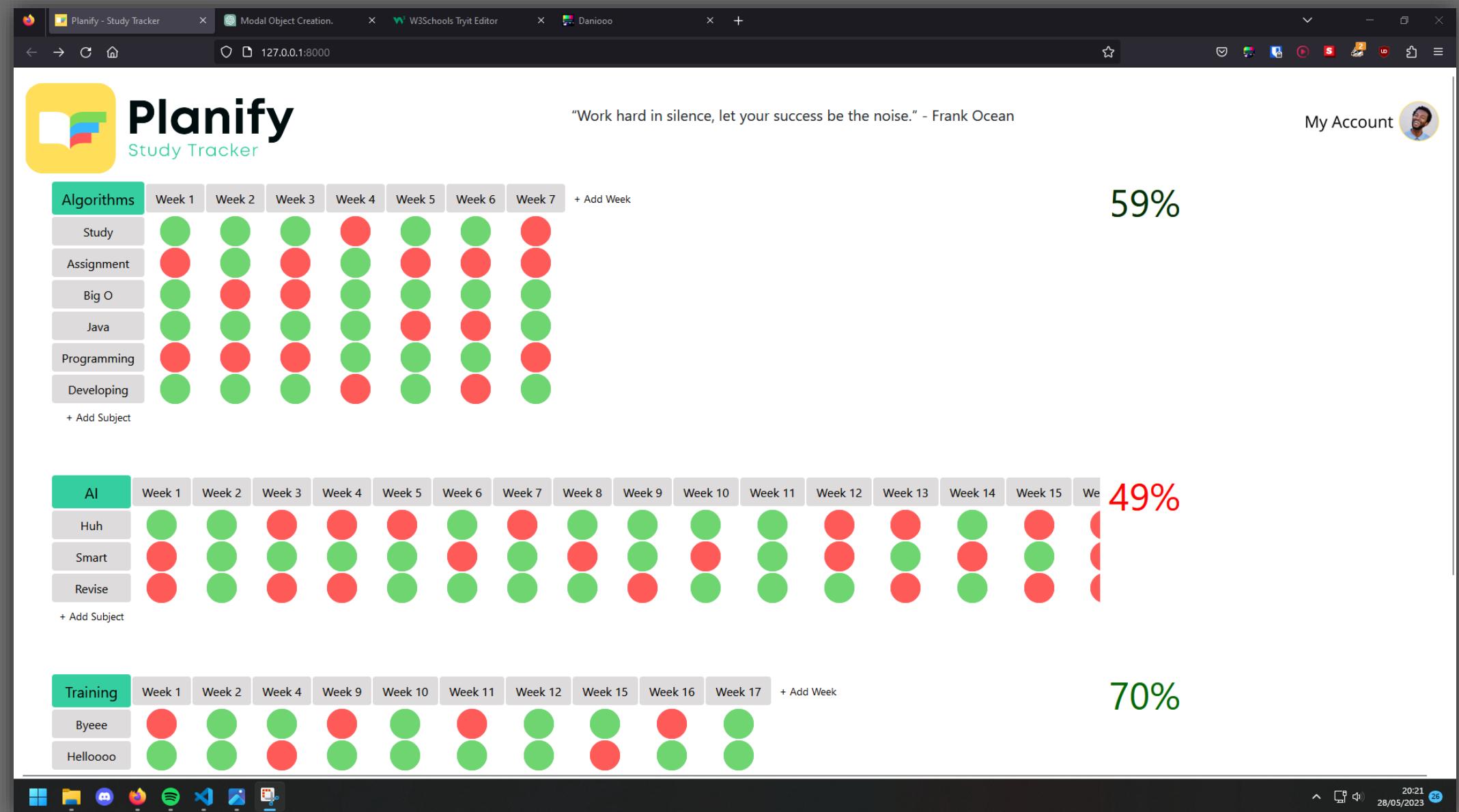
Training

	Week 1	Week 2	Week 4	Week 9	Week 10	Week 11	Week 12	Week 15	Week 16	Week 17	+ Add Week
Byeee	Red	Green	Red	Green	Green	Green	Red	Red	Red	Red	
Hellooooo	Red	Green	Red	Green	Red	Green	Red	Red	Red	Red	

99%

19:24 26  
28/05/2023

**28<sup>th</sup> May 2023 (28/05/2023)**



# 30th May 2023 (30/05/2023)

Planify - Study Tracker   Modal Object Creation.   W3Schools Tryit Editor

127.0.0.1:8000

My Account 

"Live as if you were to die tomorrow. Learn as if you were to live forever." - Mahatma Ghandi

## Algorithms

	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8
Study	Green	Green	Green	Red	Green	Green	Red	Green
Assignment	Red	Green	Red	Green	Red	Red	Red	Green
Big O	Green	Red	Red	Green	Green	Green	Green	Red
Java	Green	Green	Green	Green	Red	Red	Green	Green
Programming	Red	Red	Red	Green	Green	Red	Red	Green
Developing	Green	Red	Green	Red	Red	Green	Red	Red

+ Add Week

Likelihood of passing: 60%

## AI

	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12	Week 13	Week 14	Week 15	Week 16	Week 17	V
Huh	Green	Green	Red	Red	Green	Red	Green	Green	Green	Red	Red	Red	Green	Red	Red	Red	Red	Red
Smart	Red	Green	Red	Green	Red	Red	Red	Green	Red	Red	Red	Red	Red	Red	Red	Red	Red	Red
Revise	Red	Green	Red	Red	Green	Green	Red	Green	Red	Red	Red	Red	Red	Red	Red	Red	Red	Red

+ Add Subject

Likelihood of passing: 52%

Windows taskbar icons: File Explorer, Task View, Microsoft Edge, Firefox, Visual Studio Code, File Manager.

System tray icons: Volume, Network, Battery, Date/Time (09:12, 30/05/2023).

# 31st May 2023 (31/05/2023)

The screenshot shows a developer's environment with multiple windows open:

- Visual Studio Code (Left Window):** Displays the file `home.html`. The code includes logic for handling events like "click" on specific table rows to set them as complete or incomplete. It also handles rendering tables onto the screen.
- Browser DevTools (Top Right):** Shows the URL `127.0.0.1:8000` and a modal dialog titled "Week". The modal contains three buttons: "Set all as inactive", "Set all as complete", and "Set all as incomplete". Below the buttons are two buttons: "Delete Week" and "Close".
- Browser DevTools (Bottom Right):** Shows the "Console" tab with several log entries from the server. The log entries include:
  - [31/May/2023 09:54:29] "GET /api/getWeeks/1" 200 281
  - [31/May/2023 09:54:29] "POST /api/updateNode/41" 200 74
  - [31/May/2023 09:54:29] "POST /api/updateNode/43" 200 75
  - [31/May/2023 09:54:29] "GET /api/getNodes/1" 200 3167
- System Taskbar (Bottom):** Shows icons for File Explorer, Task View, Taskbar settings, and the date/time: 31/05/2023 09:54.

# 31st May 2023 (31/05/2023)

The screenshot shows a developer's environment with Visual Studio Code (VS Code) and a web browser.

**Visual Studio Code (Left):**

- File Explorer:** Shows files like `uris.py`, `views.py api`, `views.py base`, `signup.html`, `login.html`, `models.py`, and `home.html`.
- Code Editor:** Displays Python code for generating HTML tables based on subject data. The code includes logic for rendering subjects, nodes (white, green, red circles), and calculating a pass metric.
- Terminal:** Shows log entries from a Discord gateway connection.

**Browser (Right):**

- Planify - Study Tracker:** A web application for tracking study progress across weeks. It shows three subjects: "AI", "Huh", "Smart", and "Revise". Each subject has a row of five circles representing weeks 1 through 5. The "Huh" and "Smart" rows have one circle filled per week, while the "Revise" row has no filled circles.
- Likelihood of passing:** A large green text area displays "NaN%".
- BCS Study Tracker:** Similar to the Planify interface, it tracks progress for subjects "Saul Goodman" and "lwryup" across five weeks. Both subjects have one circle filled per week.
- Likelihood of passing:** A large green text area displays "100%".
- Console:** Shows browser developer tools with tabs for Inspector, Console, Debugger, Network, Style Editor, Performance, Memory, Storage, Accessibility, Errors, Warnings, Logs, Info, Debug, CSS, XHR, and Requests.

# 31st May 2023 (31/05/2023)

A screenshot of a Firefox browser window. The title bar shows three tabs: "Planify - Change Password", "Modal Object Creation.", and "NoReverseMatch at is not a req". The active tab's URL is "127.0.0.1:8000/account\_management/change\_password/". The page content is a password change form:

Old password\*

New password1\*

New password2\*

The browser's toolbar and taskbar are visible at the bottom, showing various icons and system status.

# 31st May 2023 (31/05/2023)

The screenshot shows the Visual Studio Code interface with the 'home.html' file open. The code is a template for a study tracker application. It includes a header with a logo, a quote, and a dropdown menu for account management. The main content area is a table for tracking subjects across weeks. The footer contains API log entries.

```
<head>
    <style>
        /* Provides horizontal scrollbar for overflowing tables */
        table {
            width: max-content;
            table-layout: auto;
            border-collapse: collapse;
            margin: 50px 30px 50px 50px;
        }
    </style>
</head>
<body>
    <div class="container-fluid">
        <div class="row">
            <div class="col-12">
                <div class="navbar-container">
                    <a href="{% url 'get-started' %}">
                        
                    </a>
                    <p id="quote-text"></p>
                    <!--<a href="{% url 'account-management' %}">
                        
                    </a-->
                    <div class="dropdown">
                        <button class="btn btn-primary dropdown-toggle" type="button" id="dropdownMenuButton" data-
                            My Account
                        </button>
                        <ul class="dropdown-menu" aria-labelledby="dropdownMenuButton">
                            <li><a href="#">Edit Account</a></li>
                            <li><a href="#">Change Password</a></li>
                            <li><a href="#">Delete Account</a></li>
                            <li><a href="#">Log Out</a></li>
                        </ul>
                    </div>
                </div>
            </div>
        </div>
    </div>
    <!-- Anchor for all tables -->
    <div id="tables-root">
        <div class="container-fluid">
            <!-- Button to display table creation modal -->
            <button id="create-table-btn" class="btn create-table-btn" data-toggle="modal" data-target="#tableCreateIn
            <!-- Modal For Table Create -->
            <div class="modal fade" id="tableCreateInteractions" tabindex="-1" role="dialog" aria-labelledby="tableCre
    </div>

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

[31/May/2023 15:25:19] "GET /api/getTable/46/ HTTP/1.1" 200 38  
[31/May/2023 15:25:19] "GET /api/getSubjects/46/ HTTP/1.1" 200 98  
[31/May/2023 15:25:19] "GET /api/getWeeks/46/ HTTP/1.1" 200 165  
[31/May/2023 15:25:19] "GET /api/getNodes/46/ HTTP/1.1" 200 609

Ln 146, Col 74 Spaces: 4 UFT-8 CRLF HTML Go Live

Reconnect to Discord Gateway

15:25 31/05/2023

The screenshot shows a web browser displaying the Planify Study Tracker dashboard. The dashboard features a grid where subjects like Algorithms, Assignment, Big O, Java, Programming, and Developing are tracked across weeks 1 through 7. Each subject has a green circle indicating a 100% likelihood of passing. A quote by Mahatma Ghandi is displayed above the grid.

Planify "Live as if you were to die tomorrow. Learn as if you were to live forever." - Mahatma Ghandi

My Account

Edit Account  
Change Password  
Delete Account  
Log Out

Algorithms	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Likelihood of passing
Study	Green	Green	Green	Grey	Green	Green	Green	Green
Assignment	Green	Green	Green	Grey	Green	Green	Green	Green
Big O	Green	Green	Green	Grey	Green	Green	Green	Green
Java	Green	Green	Green	Grey	Green	Green	Green	Green
Programming	Green	Green	Green	Grey	Green	Green	Green	Green
Developing	Green	Green	Green	Grey	Green	Green	Green	Green

+ Add Subject

100%

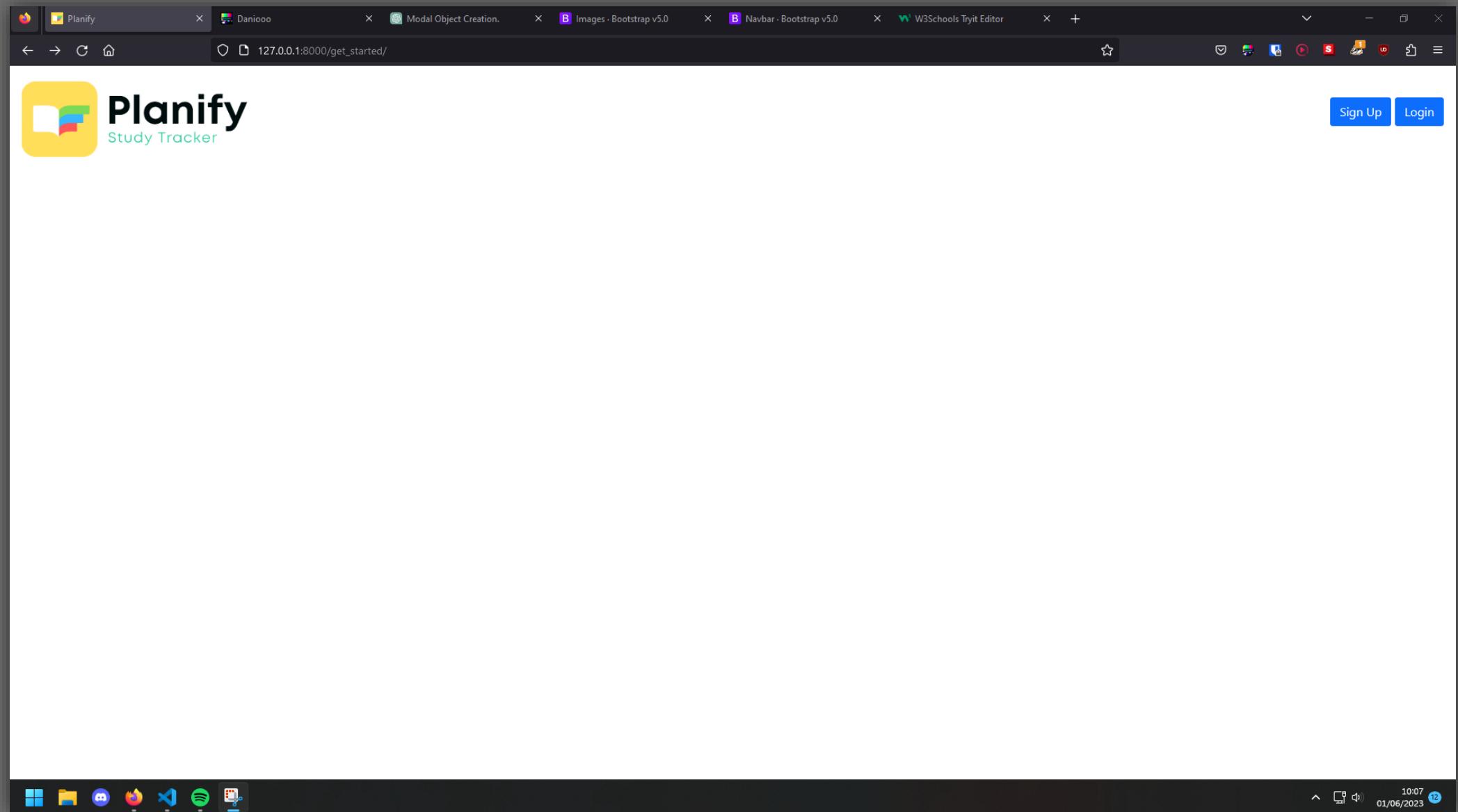
AI Week 1 Week 2 + Add Week

Huh Smart Revise + Add Subject

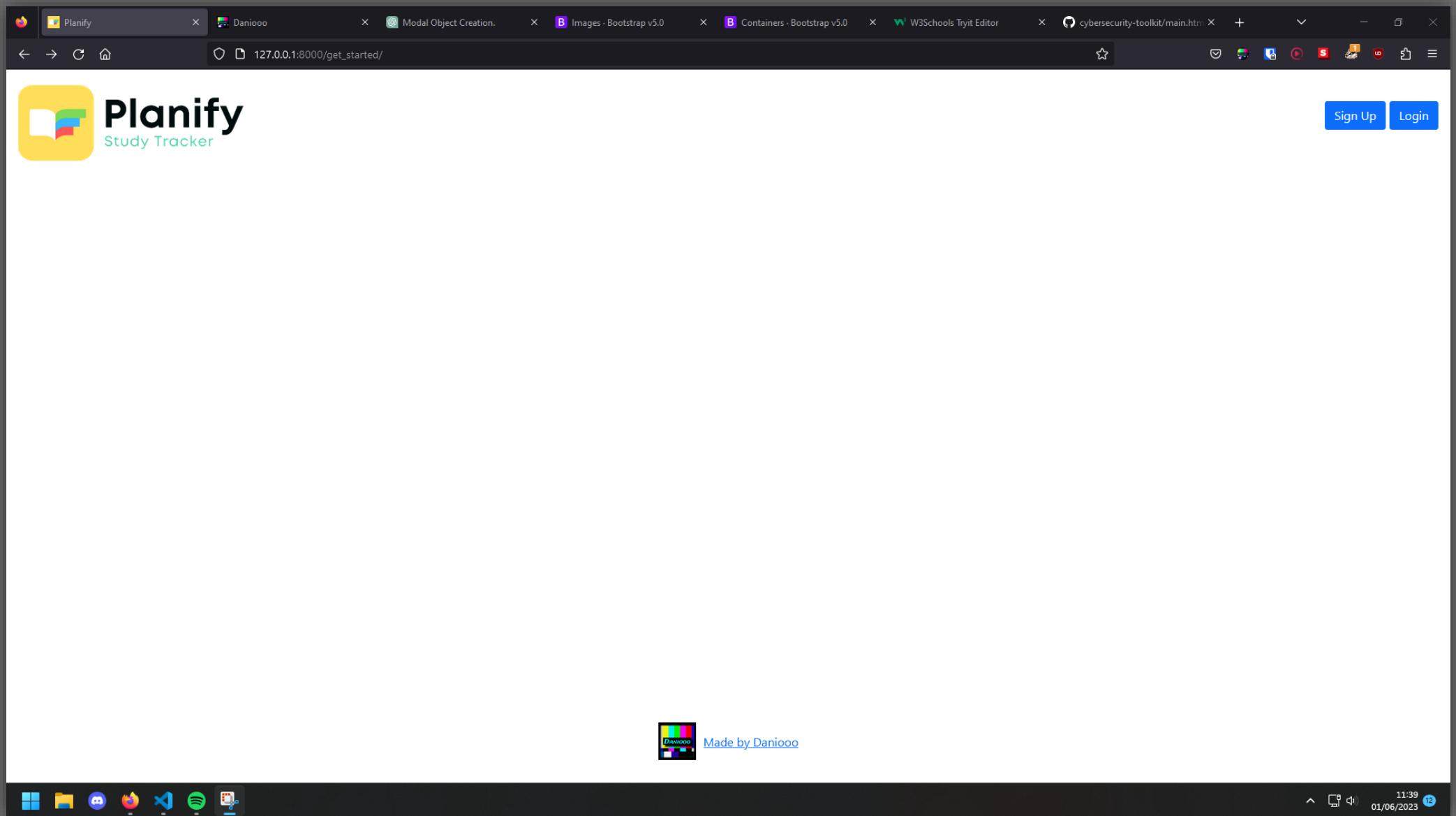
Likelihood of passing

100%

1st June 2023 (01/06/2023)



1st June 2023 (01/06/2023)



4<sup>th</sup> June 2023 (04/06/2023)

The screenshot shows a Firefox browser window with the address bar displaying "127.0.0.1:8000/signup/". The main content area shows the "Planify Study Tracker" sign-up form. The form includes fields for First name\*, Last name\*, Email\*, Username\*, and Password\*. A note below the email field states: "You will receive a code to verify your account." The browser's taskbar at the bottom shows various pinned icons, and the system tray indicates the date as 04/06/2023 and the time as 10:50.

Planify - Sign Up

Modal Object Creation.

W3Schools Tryit Editor

Forms - Bootstrap v5.0

127.0.0.1:8000/signup/

Planify Study Tracker

First name\*

Last name\*

Email\*

You will receive a code to verify your account.

Username\*

Choose a unique username

Password\*

Search

10:50 04/06/2023 13

4<sup>th</sup> June 2023 (04/06/2023)

The screenshot shows a Firefox browser window with the address bar displaying "127.0.0.1:8000/login/". The main content area features the Planify Study Tracker logo, which includes a yellow square icon with a white book and flag design, followed by the text "Planify" in large black letters and "Study Tracker" in smaller green letters. Below the logo is a light gray login form with the title "Login" at the top center. The form contains two input fields: "Username\*" and "Password\*", each with a corresponding text input box. At the bottom of the form are two buttons: a blue "Login" button and a blue "Reset Password" link. Below the form, a message reads "Don't have an account? [Sign Up](#)". At the very bottom of the page is a footer with the text "Made by Daniooo" next to a small logo consisting of colored squares.

Planify - Log X

Modal Object C W3Schools Try B Forms · Bootstrap Planify - Register https://daniooo.co 12+ Sign up Pa Postman - Sign + - ×

127.0.0.1:8000/login/

Planify Study Tracker

Login

Username\*

Password\*

Login [Reset Password](#)

Don't have an account? [Sign Up](#)

Made by Daniooo

Search

13:16 04/06/2023

4<sup>th</sup> June 2023 (04/06/2023)

The screenshot shows a Firefox browser window with the following details:

- Tab Bar:** The active tab is "Planify - Login". Other tabs include "Modal Object Creation.", "12+ Sign up Page Design", "Postman - Sign In", and "Form controls - Bootstrap".
- Address Bar:** The URL is 127.0.0.1:8000/login/.
- Content Area:** The main content is the "Planify Study Tracker" login page. It features a logo with a yellow book icon and the text "Planify Study Tracker". The login form is titled "Login" and contains fields for "Username\*" and "Password\*". Below the form are "Login" and "Reset Password" buttons, and a link "Don't have an account? [Sign Up](#)".
- Bottom Navigation:** A taskbar with icons for File Explorer, Task View, Search, and other system tools.
- System Tray:** Shows the date (04/06/2023), time (17:35), battery level (12%), and signal strength.

4<sup>th</sup> June 2023 (04/06/2023)

The screenshot shows a Firefox browser window with the address bar displaying "127.0.0.1:8000/login/". The main content area shows the Planify Study Tracker login page. The page features a yellow logo with a book and flag icon, followed by the text "Planify" and "Study Tracker". Below the logo is a large input field for "Username\*" and a smaller input field for "Password\*". A "Forgot your password?" link is provided, along with a yellow "Login" button. At the bottom, there is a link for users who don't have an account: "Don't have an account? [Sign Up](#)". The browser's taskbar at the bottom includes icons for File Explorer, Task View, Search, and several pinned applications like Postman and a CSS snippet.

4<sup>th</sup> June 2023 (04/06/2023)

Firefox | Planify - Sign Up X | Modal Object Create X | 12+ Sign up Page D X | Postman - Sign In X | B Vertical alignment · X | M justify-content - CSS X | + | - | □ | X

127.0.0.1:8000/signup/



**Planify**  
Study Tracker

Create Planify Account

First name\*

Last name\*

Email\*

You will receive a code to verify your account.

Username\*

Search

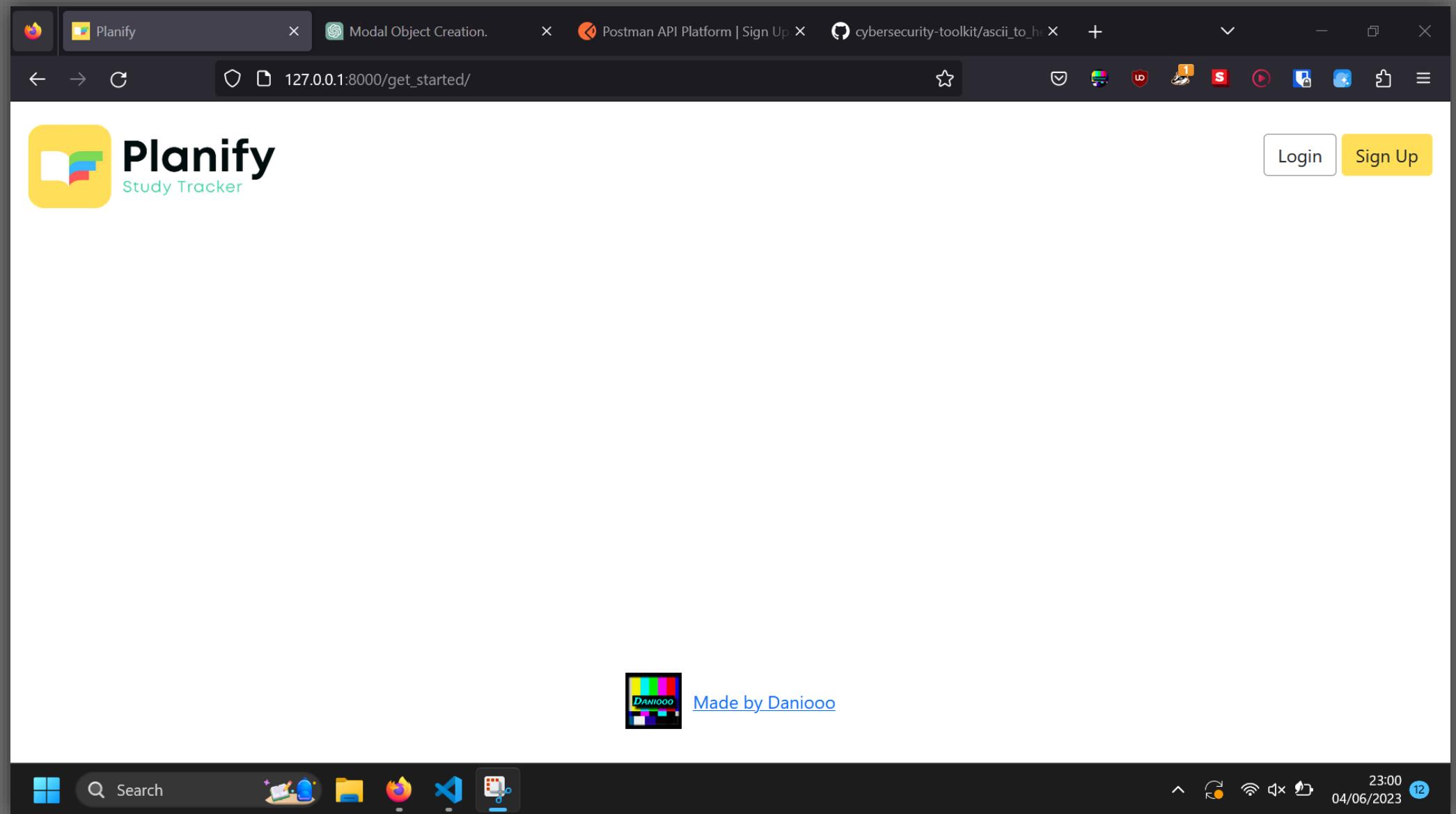
18:29 04/06/2023

4<sup>th</sup> June 2023 (04/06/2023)

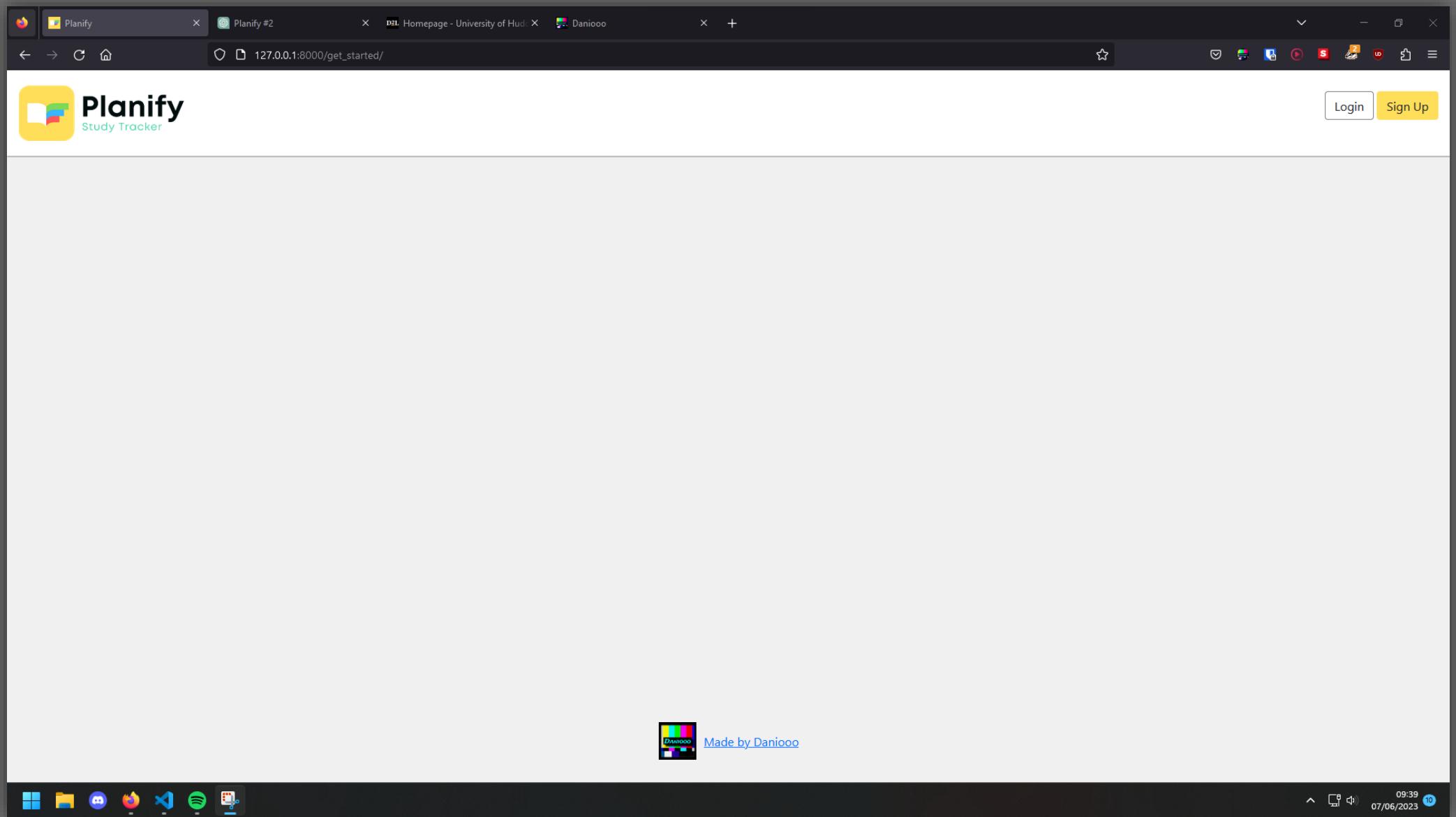
The screenshot shows a Firefox browser window with the following details:

- Tab Bar:** The active tab is "Planify - Login" at 127.0.0.1:8000/login/. Other tabs include "Modal Object Creation.", "Postman - Sign In", and "cybersecurity-toolkit/ascii\_to\_hex".
- Address Bar:** The URL 127.0.0.1:8000/login/ is displayed.
- Content Area:**
  - Logo:** The Planify logo, featuring a yellow square with a white book icon and the word "Planify" below it.
  - Title:** "Login"
  - Error Message:** A red-bordered box contains the text "Invalid username or password. Please try again."
  - Username Field:** The input field contains the text "dano00".
  - Password Field:** An empty input field for entering a password.
  - Forgot Password Link:** A link labeled "Forgot your password?" is located below the password field.
  - Login Button:** A yellow button labeled "Login" is positioned at the bottom right of the form.
- Taskbar:** The Windows taskbar at the bottom shows icons for File Explorer, Task View, and other system tools. The date and time are shown as 04/06/2023 22:50.

**4<sup>th</sup> June 2023 (04/06/2023)**



**7<sup>th</sup> June 2023 (07/06/2023)**



# 7<sup>th</sup> June 2023 (07/06/2023)

Planify - Study Tracker   Planify #2   Daniooo

127.0.0.1:8000

Planify Study Tracker

"It's no use going back to yesterday, because I was a different person then." - Lewis Carroll

My Account ▾

Likelihood of passing  
100%

Likelihood of passing  
100%

Likelihood of passing

Algorithms

	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	+ Add Week
Study	Green	Green	Green	Grey	Green	Green	Grey	Green	Green	Grey	
Assignment	Green	Green	Green	Grey	Green	Green	Grey	Green	Green	Grey	
Big O	Green	Green	Green	Grey	Green	Green	Grey	Green	Green	Grey	
Java	Green	Green	Green	Grey	Green	Green	Grey	Green	Green	Grey	
Programming	Green	Green	Green	Grey	Green	Green	Grey	Green	Green	Grey	
Developing	Green	Green	Green	Grey	Green	Green	Grey	Green	Green	Grey	

+ Add Subject

AI

	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12	Week 13	Week 14	Week 15	Week 16	Week 17	V
Huh	Grey	Grey	Grey	Grey	Grey	Grey	Grey	Grey										
Smart	Grey	Grey	Grey	Grey	Grey	Grey	Grey	Grey										
Revise	Grey	Grey	Grey	Grey	Grey	Grey	Grey	Grey										

+ Add Subject

BCS

	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	+ Add Week
--	--------	--------	--------	--------	--------	--------	------------

Likelihood of passing

10:23 07/06/2023

# 7<sup>th</sup> June 2023 (07/06/2023)

Planify - Study Tracker   ChatGPT   Daniooo   css - How do min-content and   W3Schools Tryit Editor   Homepage - University of Hud...   CSS Box Model

127.0.0.1:8000

If you fail to plan, you are planning to fail. - Benjamin Franklin

My Account ▾

Planify Study Tracker

Likelihood of passing

100%

Algorithms

	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10
Study	Green	Green	Green	Grey	Green	Green	Grey	Green	Green	Grey
Assignment	Green	Green	Green	Grey	Green	Green	Grey	Green	Green	Grey
Big O	Green	Green	Green	Grey	Green	Green	Grey	Green	Green	Grey
Java	Green	Green	Green	Grey	Green	Green	Grey	Green	Green	Grey
Programming	Green	Green	Green	Grey	Green	Green	Grey	Green	Green	Grey
Developing	Green	Green	Green	Grey	Green	Green	Grey	Green	Green	Grey

+ Add Subject

Likelihood of passing

100%

AI

	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12	Week 13	Week 14	Week 15	Week 16	Week 17
Huh	Grey	Grey	Grey	Grey	Grey	Grey	Grey										
Smart	Grey	Grey	Grey	Grey	Grey	Grey	Grey										
Revise	Grey	Grey	Grey	Grey	Grey	Grey	Grey										

+ Add Subject

Likelihood of passing

100%

BCS

	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	+ Add Week
--	--------	--------	--------	--------	--------	--------	------------

11:54 07/06/2023

**7<sup>th</sup> June 2023 (07/06/2023)**

# 7<sup>th</sup> June 2023 (07/06/2023)

Planify - Study Tracker   Planify #2   Planify - Modal - Bootstrap v5.0

127.0.0.1:8000

Planify Study Tracker

Table

Name: Algorithms

Save Changes   Delete Table   Close

Algorithms   Week 1   Week 2   Week 3   Week 4   Week 5   Week 6   Week 7   Week 8   Week 9   Week 10   Week 11   Week 12   Week 13   Week 14   Week 15   Week 16   Week 17

Study   Assignment   Big O   Java   Programming   Developing

+ Add Task

Likelihood of passing

100%

AI   Week 1   Week 2   Week 3   Week 4   Week 5   Week 6   Week 7   Week 8   Week 9   Week 10   Week 11   Week 12   Week 13   Week 14   Week 15   Week 16   Week 17

Huh   Smart   Revise

+ Add Task

Likelihood of passing

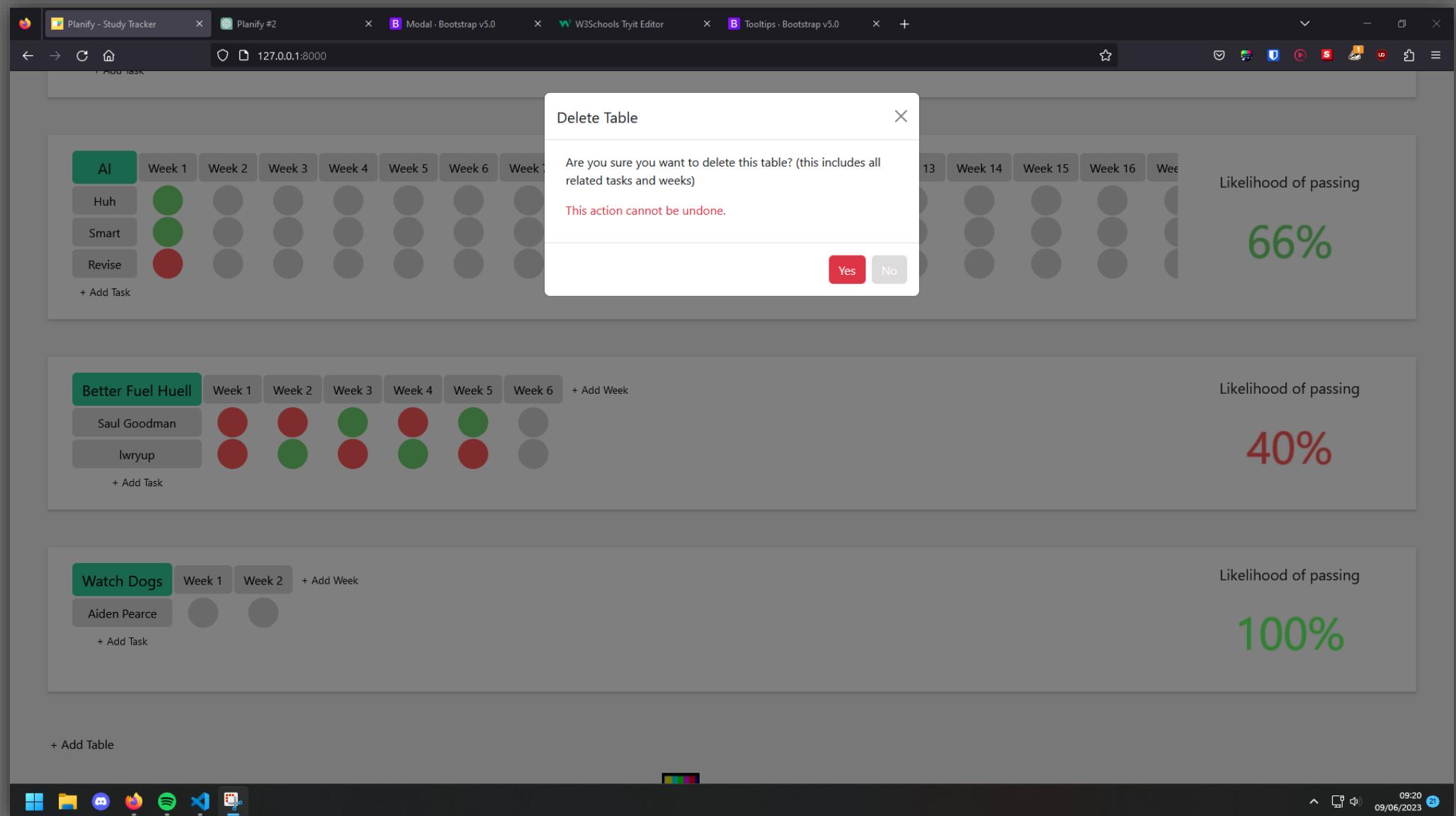
66%

BCS   Week 1   Week 2   Week 3   Week 4   Week 5   Week 6   + Add Week

Likelihood of passing

Windows Taskbar icons: File Explorer, OneDrive, Mail, Edge, Spotify, Microsoft Store, Visual Studio Code, FileZilla, 17:52, 07/06/2023

8<sup>th</sup> June 2023 (08/06/2023)



# 9<sup>th</sup> June 2023 (09/06/2023)

The screenshot shows a web browser window with multiple tabs open, displaying a study tracker application at [127.0.0.1:8000](http://127.0.0.1:8000). The application interface includes a navigation bar with icons for Home, Back, Forward, Stop, and Refresh, along with a search bar and a star icon.

The main content area displays three project sections:

- AI**: Includes a 'Huh' button, a 'Smart' button, and a 'Revise' button. A '+ Add Task' button is located below the task buttons. To the right is a 'Likelihood of passing' section showing '66%' with a green background and a red progress bar.
- Better Fuel Huell**: Includes a 'Saul Goodman' button and an 'lwryup' button. A '+ Add Task' button is located below the task buttons. To the right is a 'Likelihood of passing' section showing '50%' with a green background and a grey progress bar.
- Watch Dogs**: Includes an 'Aiden Pearce' button. A '+ Add Task' button is located below the task buttons. To the right is a 'Likelihood of passing' section showing '100%' with a green background and a green progress bar.

Below the project sections is a '+ Add Table' button. A modal window titled 'Week' is currently active over the AI project, asking 'Set all week's nodes as:' and providing three options: 'Inactive' (grey circle), 'Complete' (green circle), and 'Incomplete' (red circle). A 'Delete Week' button is also present in the modal.

The system status bar at the bottom shows various icons for file operations, battery level, signal strength, and the date/time '09/06/2023 10:13'.

# 9<sup>th</sup> June 2023 (09/06/2023)

The screenshot shows a Firefox browser window with the address bar set to 127.0.0.1:8000. The main content area displays the Planify Study Tracker homepage. On the left is the Planify logo (yellow square with a white 'P' and 'f'). In the center, a quote by Ken Poirot is displayed: "Today is your opportunity to build the tomorrow you want." Below the quote is a large, empty white space intended for creating tables. A text instruction at the bottom center says "Get started by clicking the '+ Create Table' button". At the bottom right is a green button labeled "+ Create Table". A small watermark for "Daniooo" is visible at the bottom center. The taskbar at the bottom shows various pinned icons, and the system tray indicates the date as 09/06/2023 and the time as 16:36.

Planify - Study Tracker

Planify #2

Modal - Bootstrap v5.0

W3Schools Tryit Editor

Tooltips - Bootstrap v5.0

Get Started Page designs, them

127.0.0.1:8000

My Account

Planify  
Study Tracker

"Today is your opportunity to build the tomorrow you want." - Ken Poirot

Get started by clicking the "+ Create Table" button

+ Create Table

Made by Daniooo

16:36 09/06/2023

# 9<sup>th</sup> June 2023 (09/06/2023)

The screenshot shows a web browser window with multiple tabs open at the top. The active tab is titled "Planify - Change Password" and displays the URL "127.0.0.1:8000/change\_password/". The page content is a "Change Password" form for the "Planify Study Tracker" application. The form includes fields for "Old password\*", "New password\*", "New password confirmation\*", and a "Repeat your password" field. Below the "New password" field, there are three validation rules: "Must contain at least 8 characters.", "Cannot be a commonly used password.", and "Cannot be entirely numeric.". A "Save Changes" button is located at the bottom right of the form. The browser interface includes a toolbar with various icons and a dark-themed header bar.

Planify  
Study Tracker

Return Home

Change Password

Old password\*

New password\*

- Must contain at least 8 characters.
- Cannot be a commonly used password.
- Cannot be entirely numeric.

New password confirmation\*

Repeat your password

Save Changes

Made by Daniooo

Windows taskbar icons: File Explorer, Task View, Start, Taskbar settings, File Explorer, Spotify, Visual Studio Code, Taskbar settings, Taskbar settings.

System tray icons: ENG UK, Battery, Network, Volume, 20:14, 09/06/2023, 18.

# 9<sup>th</sup> June 2023 (09/06/2023)

The screenshot shows a Microsoft Edge browser window with two tabs open. The active tab displays the Planify Study Tracker homepage at [127.0.0.1:8000/get\\_started/](http://127.0.0.1:8000/get_started/). The page features a yellow header with the Planify logo and the text "Study Tracker". On the right, there's a yellow button labeled "Manage your tables". Below the header, the main content area has a title "What is Planify?" followed by a detailed description of the product, its target user (University students aged between 18-22), and its solution (A visual and interactive tool). To the right of the description is a "Planify Demo Poster" which includes a screenshot of the software interface, a list of features like "Create A Personal Account", "Likelihood to pass module Score", and "Visible Deadlines", and a comparison table for "Planify" and "Competitors". The poster also lists sections for "Development", "Testing", "Sales and Marketing", and "Future". At the bottom of the poster, there are four circular profile pictures of team members: Adam A (CEO), Daniel K (CTO), Costa M (Creative Director), and Joel S (Senior Developer). The bottom of the browser window shows the Windows taskbar with various pinned icons.

What is Planify?

**The Problem:**  
Students underachieve or fail at university due to poor time management and motivation

**Target User:**  
University students aged between 18-22

**Our Solution:**  
A visual and interactive tool that predicts the likelihood of passing a module and highlights areas of improvement.

**Planify**

Easy to understand metrics that lets you take control of your studies

Clean UX Design

Only one glance needed to see your whole academic term

Planify is simple and fast

No Metrics available

Cluttered and complicated

Limited View

Apps are buggy and slow due to redundant content

**Competitors**

The Problem:  
Students underachieve or fail at university due to poor time management and motivation

**Daily Inspiration Quote**

**View Multiple Modules**

**Add/Customise Modules**

**Add Tasks**

**Testing:**  
Selected students will participate in a beta testing. This will allow us to see if our product is stable while also seeing how others will react to the final version.

**Sales and Marketing:**  
- Work with universities who then provide this software free to their students, similar to brightspace.  
- Onboard students directly so to bring a stronger proposition to universities when we pitch to them  
- Offer the service free to students but chargeable to education providers

**Future:**  
We want every student to surpass their potential and become the best version of themselves. We'll do this by implementing AI into our algorithms to give users accurate and helpful insights.

Adam A (CEO)  
Daniel K (CTO)  
Costa M (Creative Director)  
Joel S (Senior Developer)

Made by Daniooo

ENG UK 20:39 09/06/2023

# 9<sup>th</sup> June 2023 (09/06/2023)

The screenshot shows a web browser displaying the Planify study tracker application at [127.0.0.1:8000/get\\_started/](http://127.0.0.1:8000/get_started/). The application interface includes a navigation bar with 'Planify' logo, 'Study Tracker', 'Login', and 'Sign Up' buttons. Below the navigation is a 'What is Planify?' section with a brief description and screenshots of the application's features like 'Daily Inspiration Quote', 'View Multiple Modules', 'Add/Customise Modules', and 'Add Tasks'. To the right of this is a 'Planify Demo Poster' containing sections for 'The Problem', 'Our Solution', 'Development', 'Testing', 'Future Vision', and 'Planify Concept'. At the bottom is a 'Planify Home' dashboard showing a grid of subjects and their progress.

**What is Planify?**

It is a study tracker which me and three other team members originally envisioned back in 2021 for our team project module in the first year of university.

It offers an intuitive interface which allows the user to monitor all their subjects and track their progression throughout on a week by week basis.

On top of that, completion is measured and taken into account to determine the likelihood of success - giving the user direct feedback on their efforts.

Finally, the user is provided with motivational quotes which help inspire them to keep going and reframe their mindset positively.

**Planify's Backstory**

The main idea was cultivated by Adam A who had devised this system for himself and used it on Microsoft Excel. The concept ended up being very promising and thus we ran with it - Adam taking the role of project manager and creating the branding (logo and poster).

Me and Joel B were tasked with the technical aspect which meant we had to decide what tech stack to use, learn how to use it and then make this concept a reality. Django seemed to be most suitable for our needs and Bootstrap helped speed things up.

Our work here was split up with me dealing with all the functionality entailing managing the server and implementing all the features (back-end) while Joel made sure that the website would look how we wanted it to and customized it for a friendly user experience (front-end).

**Then and Now**

While the end result fell flat from what we anticipated, we still managed to come quite close to our ambitious goals. The user was able to create and interact with one table which was limited to 5 tasks and 12 weeks however the experience was very static as everything had to be done on pre-defined pages.

Thus we never got to execute this idea to its full extent which was due to lack of time and experience with programming. The end product lacked the flexibility that made this concept powerful and was not what we hoped for.

And so we parted our ways with this project to move onto other things in life... That is until I came back to it with all the lessons I had learnt and rebuilt it from the ground up - challenging myself to complete what I considered to be way out of my level of expertise.

What took many hours of research, planning and debugging had finally become what we had hoped to see in the first iteration of Planify.

**Planify Demo Poster**

**Planify Concept**

**Planify Home**

**System Requirements:**

- Windows 7 or later
- Mac OS X 10.6 or later
- Ubuntu 12.04 or later
- Java 8 or later
- MySQL 5.6 or later
- Apache Tomcat 7 or later
- Node.js 12.0 or later
- Cross Browser Support: Chrome, Firefox, Safari, Edge, Opera, and Internet Explorer 11+.

**Features:**

- Study Tracker: Track your subjects and modules across weeks.
- Completion Percentage: Monitor the completion of each subject/module.
- Motivational Quotes: Daily inspiration quotes to keep you motivated.
- Personal Account: Create a personal account to save progress.
- Task Management: Add and customize tasks for each subject/module.
- Deadlines: Set and track deadlines for tasks.
- Reporting: Generate reports for subjects and modules.

**Team:**

- Adam A (Project Manager)
- Joel B (Front-end)
- Christopher C (Back-end)
- James D (Design)

**Future Vision:**

We want many students to use this platform and provide them with the best possible service. Our ultimate goal is to bring a change in education and help students succeed in their studies.

**Planify Concept**

**Planify Home**

**System Requirements:**

- Windows 7 or later
- Mac OS X 10.6 or later
- Ubuntu 12.04 or later
- Java 8 or later
- MySQL 5.6 or later
- Apache Tomcat 7 or later
- Node.js 12.0 or later
- Cross Browser Support: Chrome, Firefox, Safari, Edge, Opera, and Internet Explorer 11+.

**Features:**

- Study Tracker: Track your subjects and modules across weeks.
- Completion Percentage: Monitor the completion of each subject/module.
- Motivational Quotes: Daily inspiration quotes to keep you motivated.
- Personal Account: Create a personal account to save progress.
- Task Management: Add and customize tasks for each subject/module.
- Deadlines: Set and track deadlines for tasks.
- Reporting: Generate reports for subjects and modules.

**Team:**

- Adam A (Project Manager)
- Joel B (Front-end)
- Christopher C (Back-end)
- James D (Design)

**Future Vision:**

We want many students to use this platform and provide them with the best possible service. Our ultimate goal is to bring a change in education and help students succeed in their studies.

# 9<sup>th</sup> June 2023 (09/06/2023)

Planify Study Tracker

Manage your tables

## What is Planify?

It is a study tracker which me and three other team members originally envisioned back in 2022 for our team project module in the first year of university.

It offers an intuitive interface which allows the user to monitor all their subjects and track their progression throughout on a week by week basis.

On top of that, completion is measured and taken into account to determine the likelihood of success - giving the user direct feedback on their efforts.

Finally, the user is provided with motivational quotes which help inspire them to keep going and reframe their mindset positively.

### Planify Demo Poster

**The Problem:**  
Students underachieve or fail at university due to poor time management and motivation

**Target User:**  
University students aged between 18-22

**Our Solution:**  
A visual and interactive tool that predicts the likelihood of passing a module and highlights areas of improvement.

**Development:**

- Web application as opposed to mobile application (To overcome the restricted mobile screen sizes)
- Website using HTML, CSS and some Python on backend
- Will be interactive and customisable using Javascript
- We will be doing a survey to finalise features

**Testing:**

- 1) Selected students will participate in a beta testing. This will allow us to see if our product is stable whilst also seeing how others will react to the final version.
- 2) Once we finish testing, we will transition into a full release. We'll build on the userbase of beta testers who would appreciate the full release version.

**Future Vision:**  
We want every student to surpass their potential and become the best version of themselves. We'll do this by implementing AI into our algorithms to give users accurate and helpful insights.

**Team:**

- Ashwin A. (Co-Founder) [Mobile application developer, https://github.com/daniooo/Planify-Mobile-Android.git]
- Donald K. (Co-Founder) [Frontend developer, https://github.com/daniooo/Planify-Frontend.git]
- Karan M. (Creative Director) [UI/UX designer, https://github.com/daniooo/Planify-Creative-Director.git]
- Joel B. (Senior Developer) [Backend developer, https://github.com/daniooo/Planify-Backend.git]

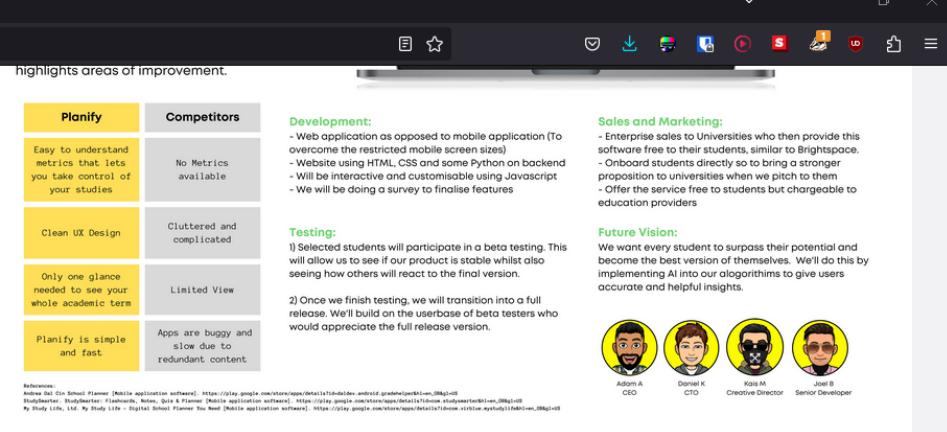
Made by Daniooo

ENG UK 22:48 09/06/2023

It offers an intuitive interface which allows the user to monitor all their subjects and track their progression throughout on a week by week basis.

On top of that, completion is measured and taken into account to determine the likelihood of success - giving the user direct feedback on their efforts.

Finally, the user is provided with motivational quotes which help inspire them to keep going and reframe their mindset positively.



**Planify**

- Easy to understand metrics that lets you take control of your studies
- Clean UX Design
- Only one glance needed to see your whole academic term
- Planify is simple and fast

**Competitors**

- No Metrics available
- Cluttered and complicated
- Limited View
- Apps are buggy and slow due to redundant content

**Development:**

- Web application as opposed to mobile application (To overcome the restricted mobile screen sizes)
- Website using HTML, CSS and some Python on backend
- Will be interactive and customizable using Javascript
- We will be doing a survey to finalize features

**Testing:**

- 1) Selected students will participate in a beta testing. This will allow us to see if our product is stable whilst also seeing how others will react to the final version.
- 2) Once we finish testing, we will transition into a full release. We'll build on the userbase of beta testers who would appreciate the full release version.

**Sales and Marketing:**

- Enterprise sales to Universities who then provide this software free to their students, similar to Brightspace.
- Onboard students directly so to bring a stronger proposition to universities when we pitch to them
- Offer the service free to students but chargeable to education providers

**Future Vision:**

We want every student to surpass their potential and become the best version of themselves. We'll do this by implementing AI into our algorithms to give users accurate and helpful insights.

**Team Photo:**



Adam A (CEO), Daniel K (CTO), Kais M (Creative Director), Joel B (Senior Developer)

**Planify Demo Poster**

**Plan To Stay Ahead**

Subject	Task	Week 1				Week 2				Week 3				Week 4				Term 1				Term 2				Term 3				Term 4			
		Wk 1	Wk 2	Wk 3	Wk 4	Wk 5	Wk 6	Wk 7	Wk 8	Wk 9	Wk 10	Wk 11	Wk 12	Wk 13	Wk 14	Wk 15	Wk 16	Wk 17	Wk 18	Wk 19	Wk 20	Wk 21	Wk 22	Wk 23	Wk 24	Wk 25	Wk 26	Wk 27	Wk 28	Wk 29	Wk 30	Wk 31	
Software Design & Development	Notes	Wk 1	Wk 2	Wk 3	Wk 4	Wk 5	Wk 6	Wk 7	Wk 8	Wk 9	Wk 10	Wk 11	Wk 12	Wk 13	Wk 14	Wk 15	Wk 16	Wk 17	Wk 18	Wk 19	Wk 20	Wk 21	Wk 22	Wk 23	Wk 24	Wk 25	Wk 26	Wk 27	Wk 28	Wk 29	Wk 30		
Computer Science & Mathematics	Notes	Wk 1	Wk 2	Wk 3	Wk 4	Wk 5	Wk 6	Wk 7	Wk 8	Wk 9	Wk 10	Wk 11	Wk 12	Wk 13	Wk 14	Wk 15	Wk 16	Wk 17	Wk 18	Wk 19	Wk 20	Wk 21	Wk 22	Wk 23	Wk 24	Wk 25	Wk 26	Wk 27	Wk 28	Wk 29	Wk 30		
Computer Organization & Architecture	Notes	Wk 1	Wk 2	Wk 3	Wk 4	Wk 5	Wk 6	Wk 7	Wk 8	Wk 9	Wk 10	Wk 11	Wk 12	Wk 13	Wk 14	Wk 15	Wk 16	Wk 17	Wk 18	Wk 19	Wk 20	Wk 21	Wk 22	Wk 23	Wk 24	Wk 25	Wk 26	Wk 27	Wk 28	Wk 29	Wk 30		

**Key**

- Completed
- Ongoing
- Low Priority
- Ignore

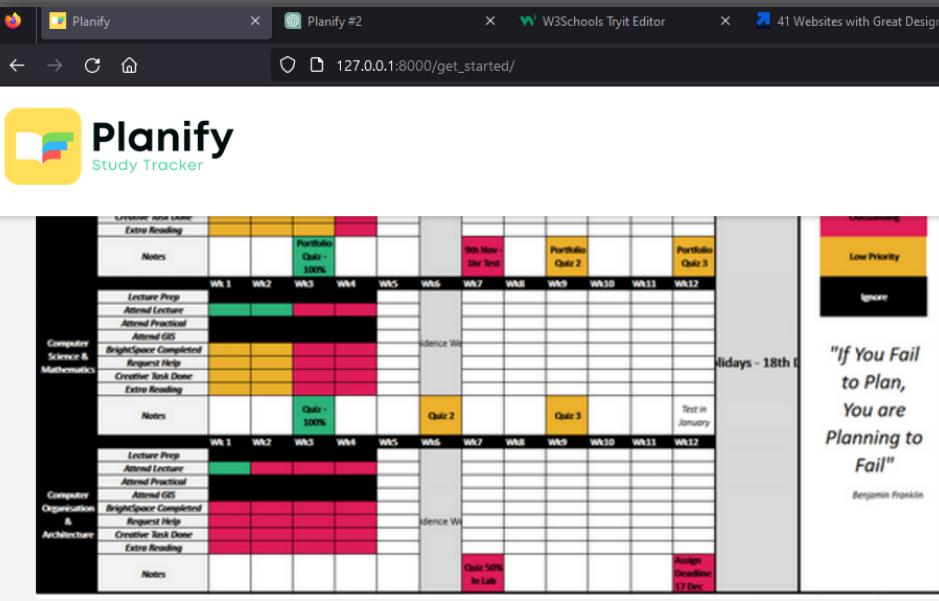
**Planify Concept**

"If You Fail to Plan, You are Planning to Fail" Benjamin Franklin



Made by Daniooo

10:00 10/06/2023



Planify Concept

Microsoft Excel. The concept ended up being very promising and thus we ran with it - Adam taking the role of project manager and creating the branding (logo and poster + user interface). Me and Joel B were tasked with the technical aspect which meant we had to decide what tech stack to use, learn how to use it and then make this concept a reality. Django seemed to be most suitable for our needs and Bootstrap helped speed things up.

Our work here was split up with me dealing with all the functionality entailing managing the server and implementing all the features (back-end) while Joel made sure that the website would look we wanted it to and customized it for a friendly user experience (front-end).

## Then vs Now

While the end result fell flat from what we anticipated, we still managed to come somewhat close to our ambitious goals. The user was able to create and interact with one table which was limited to 5 tasks and 12 weeks however the experience was very static as everything had to be done on pre-defined pages and required a lot of page hopping.

Thus we never got to execute this idea to its full extent which was due to lack of time and experience with programming. The end product lacked the flexibility that made this concept powerful and was not what we hoped for.

And so we parted our ways with this project to move onto other things in life ... That is until I came back to it with all the lessons I had learnt and rebuilt it from the ground up - challenging myself to



10:42 10/06/2023

The screenshot shows a Microsoft Edge browser window with the URL [127.0.0.1:8000/get\\_started/#section-1](http://127.0.0.1:8000/get_started/#section-1). The page displays the Planify Study Tracker logo and navigation links for "What is Planify?", "Planify's Backstory", "Then vs Now", and "Manage your tables". The main content area is titled "What is Planify?" and contains text about the product's purpose, target users, and features. It also includes a "Planify Demo Poster" section with a screenshot of the software interface and a "Planify's Backstory" section with a timeline diagram.

## What is Planify?

It is a study tracker which me (Daniel Kasprzyk) and three other team members (Adam A, Joel B and Kais M) originally envisioned back in 2022 for our team project module in the first year of university.

It offers an intuitive interface which allows the user to monitor all their subjects and track their progression throughout on a week by week basis.

On top of that, completion is measured and taken into account to determine the likelihood of success - giving the user direct feedback on their efforts.

Finally, the user is provided with motivational quotes which help inspire them to keep going and reframe their mindset positively.

## Planify Demo Poster

**Planify**

- The Problem:** Students underachieve or fail at university due to poor time management and motivation
- Target User:** University students aged between 18-22
- Our Solution:** A visual and interactive tool that predicts the likelihood of passing a module and highlights areas of improvement.

**Competitors**

Planify	Competitors
Easy to understand metrics that lets you take control of your studies	No Metrics available
Clean UX Design	Cluttered and complicated
Only one glance needed to see your whole academic term	Limited View
Planify is simple and fast	Apps are buggy and slow due to redundant content

**Development:**

- Web application as opposed to mobile application (To overcome the restricted mobile screen sizes)
- Website using HTML, CSS and some Python on backend
- Will be interactive and customizable using Javascript
- We will be doing a survey to finalise features

**Testing:**

- 1) Selected students will participate in a beta testing. This will allow us to see if our product is stable whilst also seeing how others will react to the final version.
- 2) Once we finish testing, we will transition into a full release. We'll build on the userbase of beta testers who would appreciate the full release version.

**Sales and Marketing:**

- Enterprise sales to Universities who then provide this software free to their students, similar to Brightspace
- Onboard students directly so to bring a stronger proposition to universities when we pitch to them
- Offer the service free to students but chargeable to education providers

**Future Vision:**

We want every student to surpass their potential and become the best version of themselves. We'll do this by implementing AI into our algorithms to give users accurate and helpful insights.

**Team:**

Adam A	Daniel D	Kais M	Joel B
CEO	CTO	Creative Director	Senior Developer

**Plan To Stay Ahead**

Subject	Task	Wk 1	Wk 2	Wk 3	Wk 4	Wk 5	Wk 6	Wk 7	Wk 8	Term 1	Wk 9	Wk 10	Wk 11	Wk 12	Wk 13	Wk 14	Key
Lecture Notes	Assignment	Project	Test	Quiz	Reading	Homework	Lab Report	Case Study	Group Project	Midterm Exam	Final Exam	Final Project	Final Assessment	Final Grade	Final Score	Final Report	Final Feedback

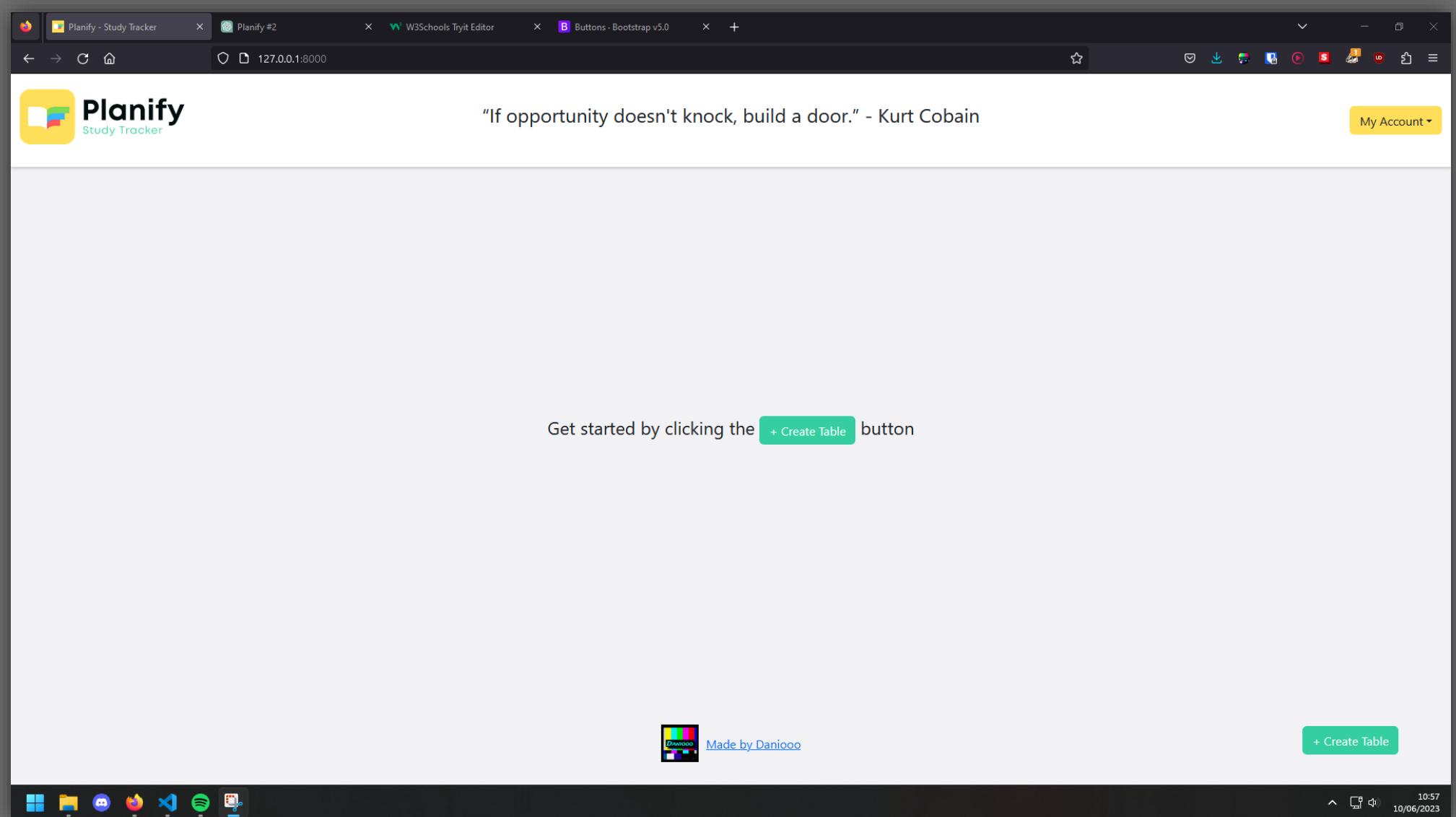
## Planify's Backstory

A Timeline Diagram illustrating the history of Planify:

- 2022: Initial idea for a study tracker is born.
- 2022: First prototype is created.
- 2022: First user feedback is gathered.
- 2022: First iteration of the software is released.
- 2023: Planify is officially launched.
- 2023: Planify is used by thousands of students worldwide.

10:52 10/06/2023

# 10<sup>th</sup> June 2023 (10/06/2023)



Planify Study Tracker

The Problem: Students underachieve or fail at university due to poor time management and motivation

Target User: University students aged between 18-22

Our Solution: A visual and interactive tool that predicts the likelihood of passing a module and highlights areas of improvement.

Planify Competitors

Planify	Competitors
Easy to understand metrics that lets you take control of your studies	No Metrics available
Clean UX Design	Cluttered and complicated
Only one glance needed to see your whole academic term	Limited View
Planify is simple and fast	Apps are buggy and slow due to redundant content

Development:

- Web application as opposed to mobile application (To overcome the restricted mobile screen sizes)
- Website using HTML, CSS and some Python on backend
- Will be interactive and customisable using Javascript
- We will be doing a survey to finalise features

Testing:

- 1) Selected students will participate in a beta testing. This will allow us to see if our product is stable whilst also seeing how others will react to the final version.
- 2) Once we finish testing, we will transition into a full release. We'll build on the userbase of beta testers who would appreciate the full release version.

Sales and Marketing:

- Enterprise sales to Universities who then provide this software free to their students, similar to Brightspace.
- Onboard students directly so to bring a stronger proposition to universities when we pitch to them
- Offer the service free to students but chargeable to education providers

Future Vision:

We want every student to surpass their potential and become the best version of themselves. We'll do this by implementing AI into our algorithms to give users accurate and helpful insights.

Team:

- Adam A - CEO
- Daniel K - CTO
- Kais M - Creative Director
- Joel B - Senior Developer

What is Planify? Planify's Backstory Then vs Now Login Sign Up

A Inspired By Failing Student Create A Personal Account Likelihood to pass module Score Add Weeks Visible Deadlines

A Inspired By Failing Student Create A Personal Account Likelihood to pass module Score Add Weeks Visible Deadlines

Sales and Marketing:

- Enterprise sales to Universities who then provide this software free to their students, similar to Brightspace.
- Onboard students directly so to bring a stronger proposition to universities when we pitch to them
- Offer the service free to students but chargeable to education providers

Future Vision:

We want every student to surpass their potential and become the best version of themselves. We'll do this by implementing AI into our algorithms to give users accurate and helpful insights.

Adam A - CEO Daniel K - CTO Kais M - Creative Director Joel B - Senior Developer

Plan To Stay Ahead

19:04 10/06/2023

# 10<sup>th</sup> June 2023 (10/06/2023)

 **Planify**  
Study Tracker

"If you fail to plan, you are planning to fail." - Benjamin Franklin

My Account ▾

Algorithms	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	+ Add Week
Study	Red	Green	Grey	Red	Green	Green	Green	Grey	Green	Red	
Assignment	Green	Green	Grey	Green	Red	Green	Green	Grey	Green	Green	
Big O	Green	Red	Grey	Green	Green	Red	Red	Grey	Green	Green	
Java	Green	Green	Grey	Red	Green	Green	Green	Grey	Red	Red	
Programming	Red	Green	Grey	Green	Green	Green	Green	Grey	Green	Green	
Developing	Green	Red	Grey	Green	Red	Green	Green	Grey	Green	Red	

Likelihood of passing **70%**

+ Add Task

AI	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12	Week 13	Week 14	Week 15	Week 16
A* Algorithm	Green	Green	Green	Green	Green	Green	Grey	Grey	Green	Green	Green	Green	Green	Green	Green	Green
Attend Lecture	Green	Green	Green	Green	Green	Green	Grey	Grey	Green	Green	Green	Green	Green	Green	Green	Green
Lab Work	Green	Green	Green	Green	Green	Green	Grey	Grey	Green	Green	Green	Green	Green	Green	Green	Green

Likelihood of passing **100%**

+ Add Task

Made by  Daniooo

+ Create Table