German University in Cairo
Faculty of Media Engineering and Technology

CSEN503 Introduction to Communication Networks
Winter Term 2022
Project Description

1. Introduction

The objective of this project is to give you an experience in developing a network application based on the client/server architecture. You are required to build a web application that is used as a simple travelling website. The website allows the users to lookup several travelling destinations. Users should be allowed to create an account, add places to their "want-to-go list" and search for travelling destinations. The website should initially be hosted on your PC's "local host" and when the site development phase is done, you should host it online on the cloud platform heroku.com.

2. Components:

- **Users Login (Main Page):**
  Registered users should be allowed to log in to their accounts using their stored username and password. If credentials are correct, the user should be redirected to the home page. If an unregistered user tries to log in an error message should be displayed.

- **User Registration:**
  Users should be allowed to create an account using a username and a password and the users' information should be stored in a database using MongoDB. If the user tried to register using an already taken username or left any of the fields empty, an error message should be displayed. After registration, the user should be redirected to the login page and a message should appear that the registration was successful.

- **Home Page:**
  The home page is the first page that should be encountered by the users when they log in to their accounts. It contains several destination categories (Beaches, Mountains, ...etc.) and a button to view the user's "want-to-go list". When the user clicks on any category, they should be redirected to that category's page.

- **Category Page:**
  The category page contains all the destinations within this category. When a user clicks on any destination's name, they should be redirected to that destination's page.

- **Destination Page:**
  The destination page contains a description for the destination. The page should also contain an embedded link for a video describing the destination which can be streamed by the user. **Please don't copy the video itself to the folder so that it doesn't exceed the allowed size**. Finally, an "add to want-to-go list" button should be added. The button adds this destination to the user's "want-to-go list" in the database. If the destination was already in the user's list, you should display an error message and don't add the duplicate destination.

- **Want-to-Go List Page:**
  The want-to-go list page contains the destinations that the user previously added using the "Add to Want-to-Go List" button. A "View Want-to-Go List" button should be added to the home page that directs the user to their own want-to-go list page.

- **Search:**
  A search bar will be displayed in all pages except for registration and login pages. The search will be done using destinations names only. The search result is either a "Destination not Found" message if the destination was not available in the database, or a list of the destinations that contain the search keyword in their names (ex: searching for "div" should put "Maldives" as one of the results). The search results should be clickable and they direct you to that specific destination's page.

3. Deployment:

After developing your web application locally on your pc, you should deploy the website on an online cloud. We will be using the following technologies for the deployment phase:

3.1 Heroku

Heroku is a cloud platform that gives the developers all of the tools and infrastructure needed to deploy their applications to a production ready server. You should create an account on heroku.com. In addition, you should download and install the Heroku Command Line Interface (CLI) to make it easy for you to create and manage your Heroku applications directly from the terminal.
You can download the Heroku CLI through the following link:
https://devcenter.heroku.com/articles/heroku-cli#download-and-install
There are different strategies used for deploying the applications on Heroku. The simplest and most common approach is to push your code from a Git repository to a Heroku app.

3.2 Git

Git is a free and open source distributed code management software. It is a version-control system for tracking changes in the source code during software development. You can download and install it through the following link:
https://git-scm.com/book/en/v2/Getting-Started-Installing-Git
To set up Git for the first time, please follow the instructions on the following link:
https://git-scm.com/book/en/v2/Getting-Started-First-Time-Git-Setup

4. Technologies:

- **Node.js:**
  Node.js is an open source platform for executing JavaScript language outside the browser (JavaScript run-time environment). You can install Node.js through the following page:
  https://nodejs.org/en/download/current/
  When the installation is done, a tool called NPM (node package manager) is also installed. NPM is used to install node packages directly through the terminal (command prompt). To know more about NPM and the available packages visit npmjs.com/

- **Express:**
  Express is a node package that is used as a web application framework. Express can be installed directly through the terminal using the command "*npm install –g express-generator"*. Express is the package responsible for running the web server.

- **Visual Studio Code (VSCode):**
  VSCode is the IDE that will be used for developing the web application. VSCode has a built-in support for JavaScript, HTML and several other languages. Furthermore, it has a built-in terminal that can be used directly to execute commands.

- **MongoDB:**
  MongoDB is a NoSQL database program. MongoDB uses JSON-like documents with optional schemas. You can download MongoDB community edition using this link (choose your operating system first):
  https://www.mongodb.com/try/download/community

  You can follow the following link for the installation steps (choose the **complete** installation option, choose to install it as a service, and keep the option to install MongoDB Compass checked [MongoDB Compass is the GUI interface for MongoDB which helps in the visualization of the database]):
  https://www.mongodb.com/docs/manual/tutorial/install-mongodb-on-windows/#install-mongodb-community-edition

- **Embedded JavaScript (EJS):**
  EJS is a template engine that allows the user to generate HTML with plain JavaScript. It allows you to load data from your application in the view. After the template is rendered, it generates an HTML file for the browser. EJS will be used as a module installed by the node package manager (NPM) along with the express server.

- **Express-Session:**
  You will use the NPM module express-session to handle multiple users logged in at the same time. Without this module users will not have their own sessions and they can access each other's data.

5. Provided Files:

You will be provided with the .ejs (HTML) files containing the basic view for your website. The files will contain plain HTML. Therefore, it is your responsibility to connect these views (frontend) with the server logic (backend) by writing the JavaScript code. You should download the zipped file as soon as it is uploaded to you and then add the provided files to your project by following the instructions provided within the zip file. There will also be VOD files that will help you setup the components for your project.

6. Submission:

Deadline: Monday, 2nd of January, 2023, 11:59 pm.

You are required to develop all the features of the website. Check the previous sections for all the details. After finishing the development phase of your application, you need to deploy it on Heroku cloud (please refer to the deployment section).

Submission Guideline: Please upload your zipped project (without the *node_modules* folder or any video files) on the following form together with the Heroku link of your website (maximum file size: 10MB):
https://forms.gle/r6gpR7873CubVzYP6

7. Resources:

- JavaScript Tutorials:
  https://www.w3resource.com/course/javascript-course.html#/
  https://www.w3schools.com/js/

- NodeJS and Server Installation:
  https://www.pluralsight.com/guides/getting-started-with-nodejs

- Creating a Simple Website Using Express:
  https://codeforgeek.com/express-nodejs-tutorial/

- HTML Basics:
  https://firstsiteguide.com/html-for-beginners/

- MongoDB with NodeJS:
  https://www.youtube.com/watch?v=_fQ_B3y3NpE&t=179s
  https://www.youtube.com/watch?v=wry-Mb2auSo

Evaluation Sheet

- Navigation (10%):
  - Handling GET requests for all pages.

- Registration (12%) [Using MongoDB is required to get the grade]:
  - Correctly getting [in the backend] the username and password provided by the user [in the frontend] (3%).
  - Checking that the username is not already in the database and that the username and password fields are not empty and displaying an error message if that was the case (one error message for both cases is enough) (5%).
  - Adding the user information correctly to the database in case the username was not already there and redirecting the user to the login page and displaying a message that the registration was successful (4%).

- Login (10%) [Using MongoDB is required to get the grade]:
  - If the user is not found in the database or entered a wrong password -> Display an error message (5%).
  - If credentials are correct -> Redirect to home page (5%).

- Multiple Users [Sessions] (7%):
  - Handling multiple users logged in at the same time (3.5%)
  - Preventing a user from accessing any page except login and registration without logging in first (3.5%).

- Videos (3%):
  - Embedding videos to destinations pages.

- Adding to Want-to-Go List (15%) [Using MongoDB is required to get the grade]:
  - If the destination is already in the user's want-to-go list -> Display an error message (7.5%).
  - If it was not -> Add it to the user's want-to-go list in the database (7.5%).

- Viewing Want-to-Go List (10%) [Using MongoDB is required to get the grade]:
  - Displaying the destinations in the current user's want-to-go list.

- Search (25%):
  - If the search key is found as a substring of a destination's name -> This destination should be displayed in the list of search results (10%).
  - The search results are clickable and can direct you to that specific destination's page (10%).
  - If the search key is not found -> Display a "Not found" message (5%).

- Deployment (8%):
  - Deploying the web application on Heroku.