

HY-359 Project

Το Project αυτό είχε ως στόχο την δημιουργία μιας ιστοσελίδας σχετικά με την πυρασφάλεια. Υπάρχουν χρήστες, εθελοντές πυροσβέστες και ένας διαχειριστής. Κάθε τύπος χρήστη έχει διαφορετικές λειτουργίες και σε κάθε χρήστη υπάρχει ασφάλεια και κάλυψη δεδομένων για τα οποία δεν έχει πρόσβαση ο χρήστης.

Κλάσεις που χρησιμοποιήθηκαν :

1. Incident { int incident_id; String incident_type, description, user_phone, user_type, address, prefecture, municipality, start_datetime, end_datetime, danger, status, finalResult; double lat, lon; int vehicles, firemen; }
2. Message {int message_id, incident_id; String message, date_time, sender, recipient;}
3. Participant {int participant_id, incident_id; String volunteer_type, volunteer_username; String status, success, comment;}
4. User {int user_id; String username, email, password, firstname, lastname, birthdate; String gender, job, afm; String country, address; Double lat, lon; String telephone; String municipality, prefecture;}
5. Volunteer {int volunteer_id; String username, email, password; String firstname, lastname, birthdate; String gender, job, afm; String country, address; Double lat, lon, height, weight; String telephone; String municipality, prefecture, volunteer_type;}

Μεθόδοι που χρησιμοποιήθηκαν :

1. Ενδεικτικά : Setters, Getters για να ορίσουμε-πάρουμε όνομα χρήστη, incident_id κλπ..
2. Ενδεικτικά : Συναρτήσεις στην js για να loadαρουμε το αντίστοιχο page, να δημιουργήσουμε δυναμικά ένα table με δεδομένα που “πήραμε” απ' την βάση κλπ.

Γενικότερη αρχιτεκτονική συστήματος :

1. Πόσα, ποια και με τι αρμοδιότητα servlets χρησιμοποιήσατε (η ότι άλλη τεχνολογία χρησιμοποιήσατε):
 - a. AdminEditUsers : Επιτρέπει στον admin να κάνει edit στοιχεία χρήστη (POST)
 - b. AllParticipants : Επιστρέφει μια λίστα με όλους τους εθελοντές (POST)
 - c. DeleteDB : Διαγραφή την βάση δεδομένων
 - d. EditParticipants : Επεξεργασία των στοιχείων εθελοντή (POST)
 - e. EmailCheck : Επιβεβαίωση αν το email υπάρχει ήδη (POST)
 - f. FindTypeServlet : Επιστρέφει τον τύπου του user. User ή Volunteer αν είναι συνδεδεμένος guest αν δεν είναι (GET)
 - g. GetStatistics : Προβάλλει τα στατιστικά που ζητούνται και δημιουργεί τα pie charts (GET)
 - h. GetUser : Επιστρέφει τα στοιχεία του συνδεδεμένου χρήστη. (GET)
 - i. IncidentsServlet : Επιστρέφει όλα τα περιστατικά με status = running. (GET)
 - j. IncidentStats : Επιστρέφει τα περιστατικά για στατιστική χρήση. (GET)
 - k. initDB : Αρχικοποίηση της βάσης
 - l. LastDateServlet : Επιστρέφει την ημερομηνία του τελευταίου μηνύματος με βάση το incident_id (GET)
 - m. Login : Είσοδος χρήστη (POST)
 - n. Logout : Έξοδος χρήστη (POST)
 - o. MessageCatalogueServlet : Επιστρέφει όλα τα μηνύματα με βάση το incident_id (POST)
 - p. NewIncident : Δημιουργία ενός νέου περιστατικού με όλα τα απαραίτητα στοιχεία στην βάση (POST)
 - q. NewMessage : Δημιουργία νέου μηνύματος για ένα περιστατικό (POST)
 - r. NotificationServlet : Έλεγχος για ειδοποιήσεις αν υπάρχουν ατυχήματα σε ακτίνα 30 χλμ απ την διεύθυνση του χρήστη (GET)
 - s. RequestToParticipate : Αίτηση ενός εθελοντή για συμμετοχή σε ένα συμβάν (POST)
 - t. TelephoneCheck : Έλεγχος διπλότυπου τηλεφώνου. (POST)
 - u. UpdateIncident : Ανανέωση στοιχείων περιστατικού (POST)
 - v. UpdateMessage : Επεξεργασία μηνύματος (POST)
 - w. UsernameCheck : Έλεγχος διπλότυπου ονόματος χρήστη (POST)
 - x. UsersServlet : Επιστρέφει η δημιουργεί νέους χρήστες (POST,GET)
 - y. VolunteersServlet : Επιστρέφει η δημιουργεί εθελοντές (POST,GET)
2. Πόσα, ποια και με τι αρμοδιότητα JavaScript Libraries/html χρησιμοποιήσατε:
 - a. DOM Manipulation : GetElementById,onload,addEventListener
 - b. Ajax Requests : Τύπος Request
 - c. Data Visualization : Pie Charts
3. Πόσα, ποια και με τι αρμοδιότητα css χρησιμοποιήσατε :
 - a. GStyles : Styling για την κύρια σελίδα
 - b. adminStyle : Styling για την σελίδα του admin
 - c. RegisterStyle : Styling για την σελίδα εγγραφής
 - d. Style : Styling για όλες τις σελίδες
 - e. UStyle : Styling για την σελίδα του χρήστη
4. Τι κάνατε στην πλευρά του client με javascript :

Δημιουργήθηκαν HTTP / AJAX requests κατά τα οποία δημιουργούσαμε έπειτα μια onload συνάρτηση και χειριζόμασταν κατάλληλα αν θα μας

επέστρεφε κωδικό 200 (ολα OK) η 400,500 (κάποιο error).Ακόμα καλούσαμε τα ανάλογα Servlets με την σωστή μέθοδο .Η GET ήταν για ανάκτηση δεδομένων απ την βάση ενώ η POST για ανανέωση η δημιουργία δεδομένων στην βάση

5. Ποια είναι τα Ajax τμήματα της εφαρμογής σας(ενδεικτικά παραδείγματα) :
 - a. AJAX για ανάκτηση δεδομένων : Ο client χρησιμοποιεί τη συνάρτηση `FindType()` για να ζητήσει τον τύπο χρήστη και το όνομα του από τον server (`FindTypeServlet`).Η απάντηση επιστρέφει σε JSON μορφή και εμφανίζεται στην οθόνη χωρίς ανανέωση της σελίδας.
 - b. AJAX για υποβολή περιστατικού : Τα δεδομένα της φόρμας υποβάλλονται στον server μέσω AJAX, χρησιμοποιώντας POST αίτημα στο `SubmitIncidentServlet`.Ο server επεξεργάζεται τα δεδομένα, τα αποθηκεύει, και επιστρέφει μήνυμα επιβεβαίωσης.
 - c. Ασύγχρονη ενημέρωση δεδομένων : Η `FindType()` τρέχει ασυγχρόνιστα και μέσω ενός promise ενημερώνονται τα δεδομένα την στιγμή που θέλουμε.
6. Που χρησιμοποιήσετε REST requests?
 - a. Ολόκληρη η αρχιτεκτονική του Project βασίζεται σε μια αρχιτεκτονική RestFul κατα την οποία, ανάκτηση δεδομένων = GET , ανανέωση δεδομένων = POST.
7. Χρησιμοποιήθηκε το OPENAI API με το οποίο υλοποιήσαμε το rolldown bar για το CHATGPT . Ακόμα χρησιμοποιήθηκε το reverse geocoding για να εξασφαλίζεται ότι όλες οι διευθύνσεις είναι πραγματικές. Τέλος χρησιμοποιήθηκε το TrueWay Matrix API με το οποίο υπολογίζεται η χιλιομετρική απόσταση απο μία διεύθυνση σε μία άλλη.
8. Αν χρειάστηκε να κάνετε αλλαγές/προσθήκες στο σχήμα της βάσης
 - a. Η βάση δεδομένων κρατήθηκε ακριβώς ίδια στην αρχιτεκτονική με την πρόσθεση μερικών παραδειγμάτων με σκοπό τον ολικό έλεγχο σωστής λειτουργίας του Project.