



**UNIVERSITY OF PATRAS  
DEPARTMENT OF ELECTRICAL ENGINEERING AND  
COMPUTER TECHNOLOGY**

**ELECTRONICS & COMPUTER SECTOR  
INTERACTIVE TECHNOLOGIES LABORATORY**

---

**Design, development and evaluation of an online  
application for planning winter sports lessons**

**THESIS**

**KONSTANTINOS STAVROPOULOS**

**SUPERVISOR: CHRISTOS SINTORIS**

**PATRAS - JULY 2025**

University of Patras, Department of Electrical Engineering and Computer Technology.

Konstantinos Stavropoulos

© 2025 – All Rights Reserved

The entire work is an original work, produced by Konstantinos Stavropoulos, and does not violate the rights of third parties in any way. If the work contains material that has not been produced by him, it is clearly visible and is explicitly mentioned in the text as the product of a third party's work, noting in a similarly clear way his identification data, while at the same time confirming that in the case of the use of unaltered graphic representations, images, graphs, etc., he has obtained the unrestricted permission of the copyright holder for the inclusion and subsequent publication of this material.

## **CERTIFICATION**

It is certified that the Diploma Thesis entitled

### **Design, development and evaluation of an online application for planning winter sports lessons**

of the student of the Department of Electrical Engineering and Computer Technology

**Konstantinos Stavropoulos of Pericles**

Registration Number: 1083868

was presented publicly at the Department of Electrical Engineering and Computer Technology on

3/7/2025

and was examined by the following committee of inquiry:

Christos Sintoris, E.D.I.P., Department of Electrical Engineering and Computer Technology (supervisor)

Sofia Daskalaki, Assistant Professor, Department of Electrical Engineering and Computer Technology (committee member)

Christos Feidas, Associate Professor, Department of Electrical Engineering and Computer Technology (committee member)

The Supervisor

Christos Sintoris  
E.D.I.P.

The Director of the Sector

Georgios Theodoridis  
Associate Professor



## **Acknowledgements**

Upon completion of my diploma thesis, I would like to express my gratitude to all those who stood by my side, supported me and guided me throughout my studies.

First of all, I would like to thank Mr. Christos Sintoris for his constant guidance and support in the preparation of my thesis. I would also like to thank Mrs. Sofia Daskalaki and Mr. Alexis Birbas for their guidance and advice during my studies. In addition, I would like to thank Mr. Nikolaos Avouris, as his lessons have been a source of inspiration for me since the first year of university.

Finally, I want to thank my family, who constantly support me, are always by my side and encourage me in my every endeavor.

## ΠΕΡΙΛΗΨΗ

### Σχεδίαση, ανάπτυξη και αξιολόγηση διαδικτυακής εφαρμογής προγραμματισμού μαθημάτων χειμερινών αθλημάτων

ΟΝΟΜΑΤΕΠΩΝΥΜΟ ΦΟΙΤΗΤΗ:

ΟΝΟΜΑΤΕΠΩΝΥΜΟ ΕΠΙΒΛΕΠΟΝΤΟΣ:

ΚΩΝΣΤΑΝΤΙΝΟΣ ΣΤΑΥΡΟΠΟΥΛΟΣ

ΧΡΗΣΤΟΣ ΣΙΝΤΟΡΗΣ

Στην παρούσα διπλωματική εργασία παρουσιάζεται η πλήρης ανάπτυξη μιας εφαρμογής για τον προγραμματισμό μαθημάτων χειμερινών αθλημάτων στα χιονοδρομικά κέντρα της Ελλάδας. Αρχικά, εξετάζονται οι υπάρχουσες λύσεις στην Ελλάδα και την Ευρώπη, από τις οποίες προκύπτει ότι υπάρχουν σημαντικές ελλείψεις — τις οποίες επιχειρεί να καλύψει η εν λόγω εφαρμογή.

Στη συνέχεια, γίνεται αναφορά στην ανθρωποκεντρική προσέγγιση σχεδίασης που εφαρμόζεται σε όλη τη διάρκεια της εργασίας. Ακολούθως, πραγματοποιείται λεπτομερής ανάλυση του προβλήματος προγραμματισμού μαθημάτων για χειμερινά αθλήματα, καθώς και ανάλυση χρηστών, εργασιών και απαιτήσεων και προσδιορίζεται το πλαίσιο PACT.

Επίσης, σχεδιάζονται αναλυτικά πρωτότυπα για τις διάφορες λειτουργίες της εφαρμογής, τα οποία αξιολογούνται από τυπικούς χρήστες.

Ακολούθως, παρουσιάζονται οι τεχνολογίες που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής, και δίνεται έμφαση στα σύγχρονα εργαλεία δημιουργίας γραφικών διεπαφών που χρησιμοποιήθηκαν, όπως η React.

Τέλος, πραγματοποιείται συνολική αξιολόγηση της εφαρμογής, προκειμένου να διαπιστωθεί σε ποιο βαθμό επιτεύχθηκαν οι αρχικοί στόχοι. Από την αξιολόγηση προκύπτει ότι η εφαρμογή ανταποκρίνεται πλήρως στις απαιτήσεις των χρηστών, προσφέροντας μια εξαιρετική εμπειρία χρήστη.

## **EXTENSIVE ENGLISH SUMMARY**

### **Design, development and evaluation of an online application for planning winter sports lessons**

**STUDENT NAME, SURNAME:**

**KONSTANTINOS STAVROPOULOS**

**SUPERVISOR NAME, SURNAME:**

**CHRISTOS SINTORIS**

This thesis presents the full-stack development of an application for planning winter sports lessons at ski resorts across Greece. Initially, existing solutions in both Greece and Europe are examined, revealing significant gaps that the proposed application aims to address.

The work adopts a human-centered design approach, which is applied throughout the development process. Moreover, a thorough analysis of the lesson scheduling problem for winter sports is performed, along with user analysis, task analysis, and requirement analysis and the PACT framework is defined.

Furthermore, detailed prototypes are developed for the various features of the application and are evaluated by representative users.

Additionally, the various technologies used for the implementation are presented, with emphasis on the modern front-end development tools that were used, such as React.

Finally, a comprehensive evaluation of the application is carried out to assess the extent to which the initial goals were achieved. The results indicate that the application successfully meets user requirements, delivering a high-quality user experience.



## Contents

1	Introduction.....	19
1.1	Purpose and motivation .....	19
1.2	Chapter structure .....	19
2	Human-centric design and current solutions.....	21
2.1	Study of applications in Greece .....	21
2.1.1	Snowport School.....	22
2.1.2	Independent instructors of Parnassos Ski Center.....	23
2.1.3	4SeasonsOutDoor School.....	24
2.1.4	Ski Center 3-5 Pigadia .....	25
2.2	Study of applications in Europe .....	26
2.2.1	Masonic Sport App.....	27
2.2.2	CheckYeti App .....	29
2.2.3	Freedom Snowsports App.....	30
2.3	Human-centered design.....	31
2.3.1	Usability .....	31
2.3.2	User Experience .....	32
2.3.3	Principles of human-centered design .....	33
2.3.4	Asteroid model.....	34
2.4	Conclusions.....	34
3	Methodology .....	37
3.1	Problem Analysis .....	37
3.2	App design .....	38
3.3	Code implementation .....	38
3.4	Application Evaluation .....	39

4	Analysis.....	41
4.1	PACT Framework - Theory.....	41
4.2	User Analysis .....	42
4.2.1	Stakeholders .....	42
4.2.2	Personas.....	42
4.3	Requirements Analysis .....	47
4.4	Hierarchical analysis.....	50
4.5	PACT Usage Framework - Results .....	53
4.5.1	Scope of use .....	53
4.5.2	PACT Framework - Results .....	54
4.6	Conclusions.....	55
5	Design.....	57
5.1	Importance of design .....	57
5.2	Design principles .....	58
5.3	Prototype design .....	59
5.3.1	Home screen .....	59
5.3.2	Login to the app .....	59
5.3.3	Signup.....	60
5.3.4	Selecting Course Parameters .....	61
5.3.5	Selection of courses .....	62
5.3.6	View instructor profiles .....	64
5.3.7	View Course Hours.....	65
5.3.8	Overview of selected courses .....	66
5.3.9	Payment .....	66
5.3.10	Overview of Student Scheduled Courses .....	66

5.3.11	Student profile .....	69
5.3.12	Managing Coach Courses .....	69
5.3.13	Instructor program.....	71
5.3.14	Instructor profile .....	72
5.3.15	Instructor statistics.....	73
5.4	Evaluating prototypes with users.....	74
5.4.1	Think aloud protocol.....	74
5.5	Conclusions.....	76
6	Implementation.....	77
6.1	Architecture.....	77
6.2	Customer-Side Technologies.....	78
6.2.1	Modern User Interface Development Technologies .....	78
6.2.2	React.js .....	79
6.2.3	Vite .....	87
6.2.4	Internationalization (i18next) .....	88
6.2.5	Charts ( Chart.js) .....	89
6.2.6	CSS.....	90
6.3	Server-side technologies .....	100
6.3.1	Node.js .....	100
6.3.2	Express.js.....	100
6.3.3	Data validation .....	101
6.3.4	EmailJS.....	101
6.3.5	Safety (Helmet) .....	101
6.4	Database.....	102
6.4.1	Entity-relationship diagram .....	102

6.4.2	PostgreSQL.....	103
6.4.3	Indexes .....	103
6.4.4	User password protection .....	104
6.5	Implementation time required .....	104
6.6	Conclusions.....	105
7	Evaluation.....	107
7.1	Methods used.....	107
7.2	Expert Evaluation .....	107
7.2.1	Heuristic evaluation .....	107
7.2.2	Cognitive Jogthrough .....	112
7.3	User evaluation .....	114
7.3.1	Thinking aloud Protocol .....	114
7.3.2	Interviews.....	115
7.3.3	UEQ short questionnaire .....	117
7.4	Changes .....	119
7.5	Conclusions.....	120
8	Conclusions.....	121
8.1	Results .....	121
8.2	Summary .....	121
8.3	Future steps.....	122
9	Bibliography .....	123

## List of Tables

Table 1. Today's solutions for scheduling lessons in ski resorts in Greece. ....	21
Table 2. Today's solutions for scheduling lessons in Europe's ski resorts. ....	26
Table 3. Table of functional requirements. ....	47
Table 4. Table of non-functional requirements. ....	49
Table 5. Results of Cognitive Jogthrough for Lesson Planning. ....	113
Table 6. Results of Cognitive Jogthrough for the supervision of a coach program. ....	113
Table 7. Characteristics of users who participated in the assessment using the method of interviews and the Thinking Aloud Protocol. ....	114
Table 8. Scenarios performed by users during the assessment using the method of interviews and the Talking Subject Protocol. ....	115
Table 9. Findings of the evaluation with users using the method of interviews and the Talking Subject Protocol. ....	116
Table 10. UEQ short questionnaire results. ....	118
Table 11. Aggregate short UEQ results. ....	118



## Image Catalog

Image 1. Snowport School.....	22
Image 2. Criteria for the selection of coaches.....	23
Image 3. Independent ski coaches at Parnassos Ski Resort. ....	23
Image 4. 4seasonsoutdoor school. ....	24
Image 5. Possibility to book courses at the 4SeasonsOutDoor school.....	25
Image 6. Possibility to book lessons at the Ski Center 3-5 Pigadia.....	25
Image 7. The Maison Sport app.....	27
Image 8. Course parameter selection form.....	28
Image 9. Coaches available for customer's preferences. ....	28
Image 10. checkYeti app. ....	29
Image 11. Ability to schedule lessons in the checkYeti app. ....	30
Image 12. Course planning at the Freedom Snowsports school. ....	30
Image 13. Possibility to book a course at the Freedom Snowsports school. ....	31
Image 14. UX Honeycomb v. (Morville,2016).....	33
Image 15. Asteroid model (Hartson & Hix, 1986).....	34
Image 16. Methodology that will be applied in the implementation of the diplomacy. ....	37
Image 17. Persona 2: Mary Papadatou.....	43
Image 18. Persona 3: Dimitris Athanasiou.....	44
Image 19. Persona 4: Michael Zerbas.....	45
Image 20. Persona 5: Artemis Dimelli.....	46
Image 21. Hierarchical analysis of tasks for course planning. ....	51
Image 22. Hierarchical analysis of tasks for the creation of a lesson by a coach.....	52
Image 23. Hierarchical analysis for program oversight. ....	53
Image 24. Easy Snow app home screen.....	59

Image 25. User login to the app.....	60
Image 26. Select a membership.....	61
Image 27. Register a new user.....	61
Image 28. Lesson planning.....	62
Image 29. Course selection, top of page. ....	63
Image 30. Course selection, bottom of page.....	64
Image 31. Coach profile. ....	65
Image 32. View available courses. ....	66
Image 33. Overview of selected courses. ....	67
Image 34. Payment menu.....	68
Image 35. Overview of ski lessons. ....	68
Image 36. Skier profile. ....	69
Image 37. Coach Course Management, Top Page.....	70
Image 38. Manage Coach Courses, Bottom Section of Page.....	71
Image 39. Coach program.....	72
Image 40. Excerpt from the coach's profile.....	73
Image 41. Coaching statistics.....	73
Image 42. Redesign user profiles.....	75
Image 43. Redesign a payment page.....	75
Image 44. Easy Snow application architecture.....	77
Image 45. Asynchronous loading of reviews. ....	82
Image 46. Asynchronous loading of courses. ....	82
Image 47. Change UI when navigating to a different page. ....	83
Image 48. The selected page has a different color in the menu. ....	84
Image 49. Update a successful email send. ....	85

Image 50. Update of successful review submission.	85
Image 51. Multi-criteria course selection menu.	87
Image 52. Vite configuration.	88
Image 53. Coach statistics.	89
Image 54. Mobile Home Screen.	91
Image 55. Mobile user reviews.	92
Image 56. User reviews on tablets.	93
Image 57. Mobile registration page.	94
Image 58. Choice of courses on tablets.	95
Image 59. Choice of courses on mobile.	95
Image 60. Mobile payment page.	96
Image 61. Create a coach teaching on a tablet.	97
Image 62. Create a coach teaching on mobile.	98
Image 63. Coach program page on tablet.	99
Image 64. Coach program page on mobile.	99
Image 65. Entity-correlation diagram of the Easy Snow application.	103
Image 66. Informing user about incorrect input fields.	109
Image 67. White space and soft colors in the Easy Snow app.	111
Image 68. Explanation of necessary code characteristics.	112
Image 69. UEQ short questionnaire.	117
Image 70. Initial calendar implementation where the user can select all dates.	119
Image 71. Improved calendar implementation where the user is shown only the dates when there are available courses.	120



# 1 Introduction

## 1.1 Purpose and motivation

Many young or even old people choose to learn a winter sport, such as skiing or snowboarding, for the first time in their lives. However, many do not know how to find an available coach, so first they go to the mountain and look for a coach there and in case there is no one, they simply try to learn on their own for a while without usually succeeding. Also, many people want to schedule a lesson several days before going to the mountain but do not know which coach to choose, on the one hand because there is little information on the internet about available coaches, who usually belong to a specific school, and on the other hand because there is no information that helps them to distinguish a coach and choose the one they consider best.

In this context, it is understood that there is a clear lack of information on the organization, finding and planning of courses with coaches who teach winter sports, while the demand is very high. Therefore, the purpose of this thesis is to design, develop and evaluate an application that enables winter sports coaches, regardless of whether they belong to a school or not, to offer lessons at any ski resort in Greece and at the same time allow skiers to choose the coaches that best meet their needs.

Besides, another important reason why this topic was chosen as a diplomatic topic is the author's long-term experience with skiing, as it is one of his favorite sports.

Finally, another pursuit of this thesis is the familiarization with modern technologies for the development of user interfaces, such as React.

## 1.2 Chapter structure

**Chapter 2** first presents the solutions that exist today for the scheduling of courses in ski resorts in Greece and Europe. Then the anthropocentric design theory is analyzed, which forms the basis for the rest of the work.

**Chapter 3** briefly presents the methodology to be applied in the following chapters and indicates which techniques will be used in each process.

**Chapter 4** analyses the problem of course scheduling. In fact, the theoretical background for the PACT framework is first defined. Subsequently, a user analysis is carried out to determine the typical characteristics of future users of the application. Furthermore, a requirements analysis is carried out and the requirements to be met by the application are defined. follow users to perform typical actions. Finally, the analysis that is carried out determines the individual elements of the PACT framework.

In **Chapter 5**, prototypes of the application are created. Specifically, after the requirements have been analyzed based on the principles of human-centered design, detailed prototypes are produced for all individual functions. Furthermore, according to human-centered theory, prototypes are evaluated with user participation in order to correct possible issues.

In **Chapter 6**, the implementation of the application in a program is carried out. In fact, the architecture of the application is analyzed and the choices that have been made regarding the technology to be used are explained. Of course, the technologies on the client-browser side, on the server side and on the database are analyzed separately.

**Chapter 7** presents the evaluation methods that were applied after the development of the application. In fact, the evaluation was carried out by both experts and users and the results obtained are recorded in detail.

## 2 Human-centric design and current solutions

The first step in design is to study related applications to understand which capabilities are covered by existing solutions and which are not. To this end, we will first examine how the problem is solved today in Greece and then we will study applications used in major ski resorts throughout Europe. This distinction is made on the one hand because we want to initially build an application for Greek ski resorts, so we need to know the situation that exists today, and on the other hand, because it is expected that the solutions in the major ski resorts of Europe will be better than the current solutions in Greece today, so they may be a source of inspiration for our application. Finally, in this chapter we review the basic principles of anthropocentric design, which will determine the actions of the following chapters.

### 2.1 Study of applications in Greece

Initially, the solutions in Greece are being studied in order to make an overview of the current situation. In fact, in Greece, usually all winter sports schools have an electronic presence and the interested person can be informed about basic things such as the courses offered, the packages and the names of the school's coaches. Furthermore, those who wish to schedule a course can contact the school's secretariat by phone. However, Table 1 shows all the applications studied in Greece. In fact, these schools and websites with independent coaches were chosen because they are the most important solutions that exist today in Greece for the planning of courses in ski resorts.

Of course, it is emphasized that the following images have been blurred in some places for privacy reasons, such as the phone, name or email address.

*Table 1. Today's solutions for scheduling lessons in ski resorts in Greece.*

n/a	Appellation	Features
1	Snowport School <sup>1</sup>	The names of the coaches, the sport each teaches and brief biographical information are provided. Course scheduling is done by phone.
2	Coaches at Parnassos Ski Center <sup>2</sup>	The name and contact details of various independent coaches are provided.
3	4SeasonsOutDoor School <sup>3</sup>	The school's phone number, location, email address, and course cost information are provided.

<sup>1</sup> <https://www.snowport.gr/ski-school>

<sup>2</sup> <https://parnassos-ski.gr/episkepsi/mathimata>

<sup>3</sup> <https://4seasonsoutdoor.gr/kratiseis>

n/a	Appellation	Features
4	Coaches at the ski resort 3-5 Pigadia <sup>1</sup>	A contact form is provided that the interested party must fill in regarding the hours the course wishes to last, the type of sport, the level of the user, how many will participate and the date. The lesson is scheduled at a later time.

### 2.1.1 Snowport School

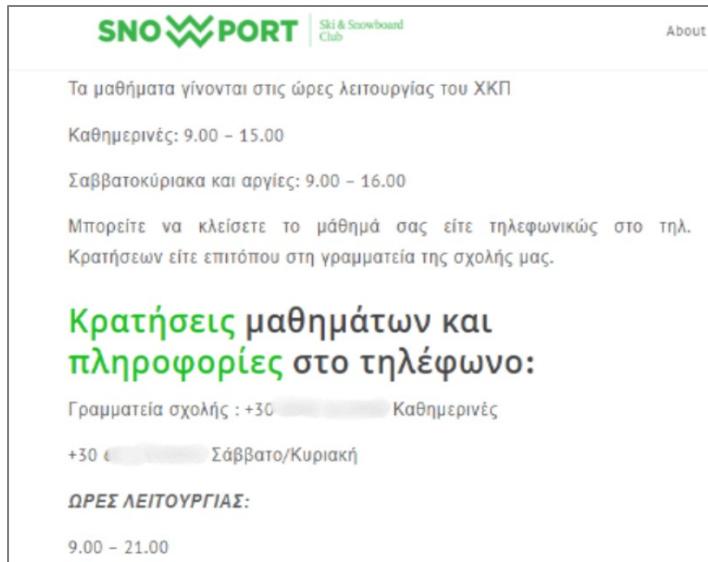


Image 1. Snowport School.

The Snowport [School](#)<sup>2</sup> is based in the Parnassos Ski Centre, which is the largest in Greece, and provides ski and snowboarding lessons for all ages. Above is the page that the school has to plan a lesson. As shown in figure 1, the interested person is informed by the website about the school's telephone number and can contact the secretariat to request additional information or to schedule a lesson. Also, he can be informed about the coaches of the school and separate them according to the sports that each one teaches, as shown in figure 2. In fact, for each coach, the interested person can read his CV and contact details.

<sup>1</sup> <https://35pigadia.com/scholi-ski-snowboard>

<sup>2</sup> <https://www.snowport.gr/ski-school>

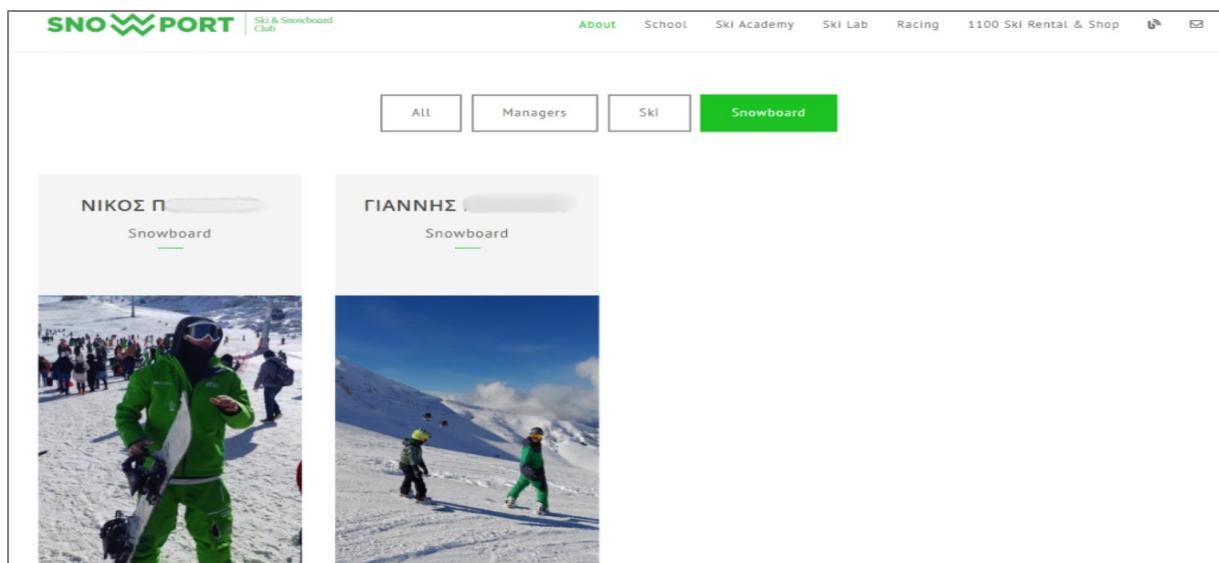


Image 2. Criteria for the selection of coaches.

### 2.1.2 Independent instructors of Parnassos Ski Center

The screenshot shows the Parnassos Ski Center website's coaching section. At the top, it says 'ΣΤΕΛΙΟΣ' (Stefanos). Below it, it says 'Εκπαιδευτής Χιονοδρομίας | Συναδός Βουνού | Παιδαγωγός'. A list of criteria follows: '• Ενεργό μέλος, Πανελλήνιας Ένωσης Προπονητών-Εκπαιδευτών Χιονοδρομίας www...' and '• Ενεργό μέλος, Ελληνικού σωματείου αρχηγών βουνού, hn...'. Below this is a photo of Stefanos, a man in a dark blue snowsuit with a logo, standing on skis. To the right is a contact card with a phone icon and the number 69, followed by a large redacted area. Below Stefanos' photo is another section for 'ΑΝΔΡΕΑΣ'. It says 'Καθηγητής Φυσικής Αγωγής' and 'Προπονητής Χιονοδρομίας – Εκπαιδευτής Ski-Snowboard'. It includes a photo of Andreas, a man in a green snowsuit with two children, and a contact card with icons for phone (69), WhatsApp, Vodafone, email (andreas@parnassos-ski.gr), and Facebook (an).

Image 3. Independent ski coaches at Parnassos Ski Resort.

Parnassos Ski Center<sup>1</sup> provides a lot of useful information for future visitors on its website. One of them is the planning of ski lessons and allows independent coaches, i.e. people who do not

<sup>1</sup> <https://parnassos-ski.gr/episkepsi/mathimata>

belong to a school, to be promoted and provide their contact details. Of course, these details are usually very limited, such as the coach's phone or email address. As most coaches do not have a personal website to refer the interested person.

### 2.1.3 4SeasonsOutDoor School

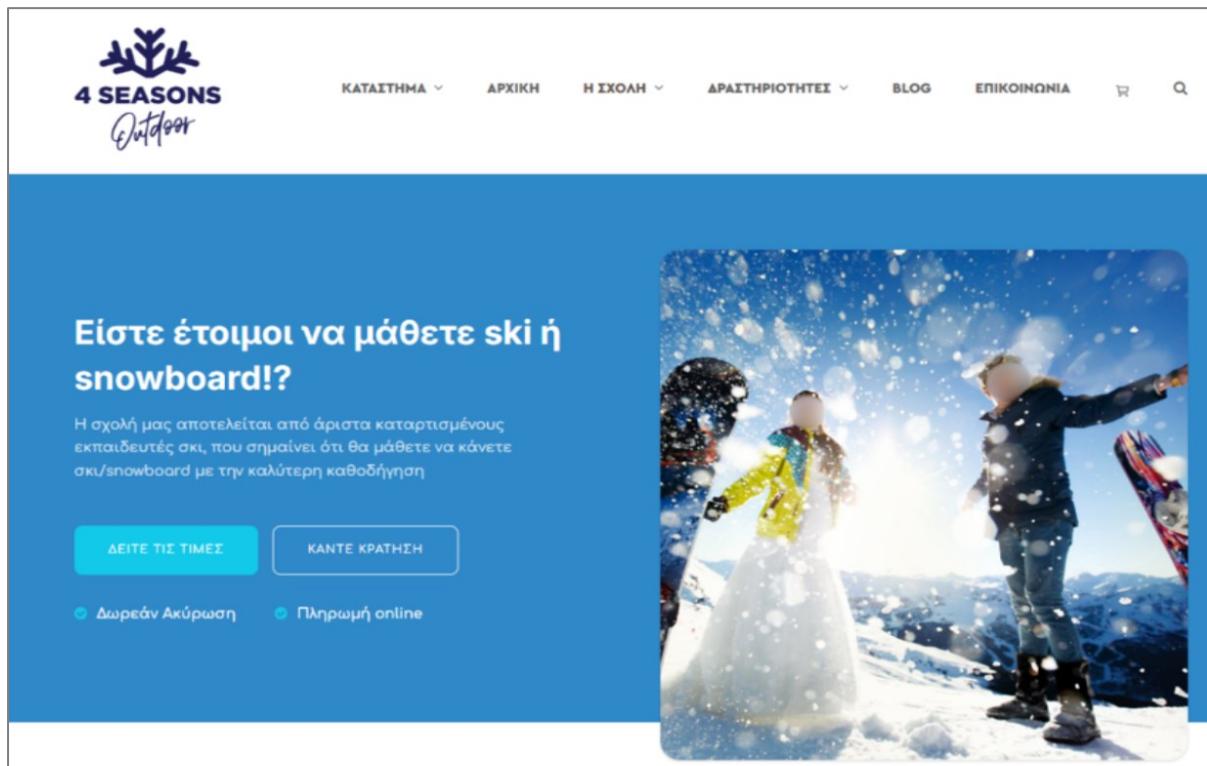


Image 4. 4seasonsoutdoor school.

The 4SeasonsOutDoor [school](#)<sup>1</sup> is active in the Kalavryta Ski Center and provides ski and snowboarding lessons. Figure 4 shows the website that exists for users who wish to schedule a lesson. In addition, users can find out about course prices and equipment required. In fact, if the interested parties click on the "Make a Reservation" option, then they are redirected to the window of image 5, which informs them about the school's phone, its location and the e-mail address.

<sup>1</sup> <https://4seasonsoutdoor.gr/kratiseis>

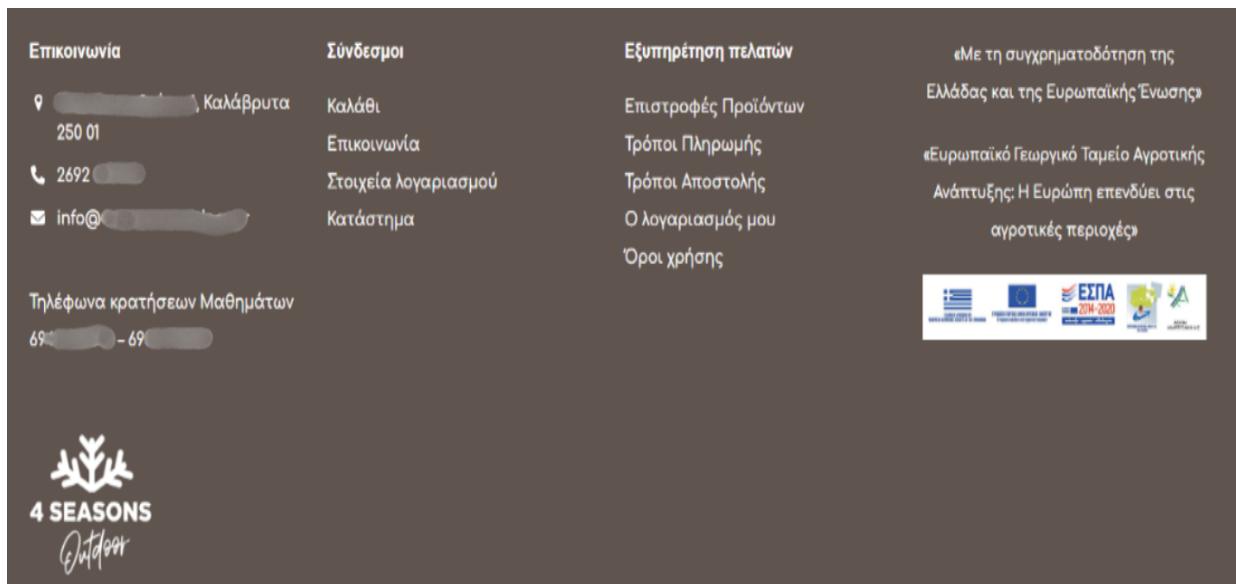


Image 5. Possibility to book courses at the 4SeasonsOutDoor school.

#### 2.1.4 Ski Center 3-5 Pigadia

The screenshot shows a booking form for ski/snowboard lessons. It includes fields for name, email, phone number, number of students (1-5), number of hours (1-4), lesson type (Ski/Snowboard), level (Beginner/Middle/Advanced), date, and a checkbox for booking multiple hours. A 'Send' button is at the bottom.

Image 6. Possibility to book lessons at the Ski Center 3-5 Pigadia.

One of the small ski resorts in Greece is [3-5 Pigadia](#)<sup>1</sup>, which is located in Naoussa. The website of this ski resort gives the visitor the opportunity to schedule lessons, as shown in figure 6. In fact, the interested party can set the hours he wishes to last the lesson, the type of sport, the level of competence of the student, How many will participate and the date. Of course, the visitor is not informed about the available coaches nor about the days and times when classes are held.

<sup>1</sup> <https://35pigadia.com/scholi-ski-snowboard>

Therefore, initially the visitor must submit the above form for the expression of interest and at a later time schedule the course.

## 2.2 Study of applications in Europe

In this section we study the applications in the major ski resorts of Europe as they are likely to have some interesting features from which we may be inspired for our own application. In fact, these applications offer users features that vary per application and we mention them in Table 2, where the applications studied are shown in aggregate. Of course, it is emphasized that the following images have been blurred in some places for reasons of protection of the privacy of the people depicted.

*Table 2. Today's solutions for scheduling lessons in Europe's ski resorts.*

n/a	Appellation	Features
1	Maison Sport App <sup>1</sup>	The user when registering in the application chooses whether he is a coach or a student. Then, if they are a student, they choose the ski resort they are interested in, the desired dates and the sport, such as regular skiing and skiing for disabled people. Also, the user has access to reviews of other users.
2	CheckYeti App <sup>2</sup>	Information is provided on a variety of sports such as windsurfing, skiing, snowboarding, etc. Of course, the user is not given the opportunity to be informed about who the coaches are in each lesson.
3	Freedom Snowsports School <sup>3</sup>	The user is given the opportunity to choose a course according to the desired sport, his age, his skill level and his time preferences. Of course, the available coaches are quite limited as they are those who belong to this particular school.

---

<sup>1</sup> <https://maisonsport.com/en/resort>

<sup>2</sup> <https://www.checkyeti.com/en/v/1/winter>

<sup>3</sup> <https://www.book.ski/team>

## 2.2.1 Masonic Sport App

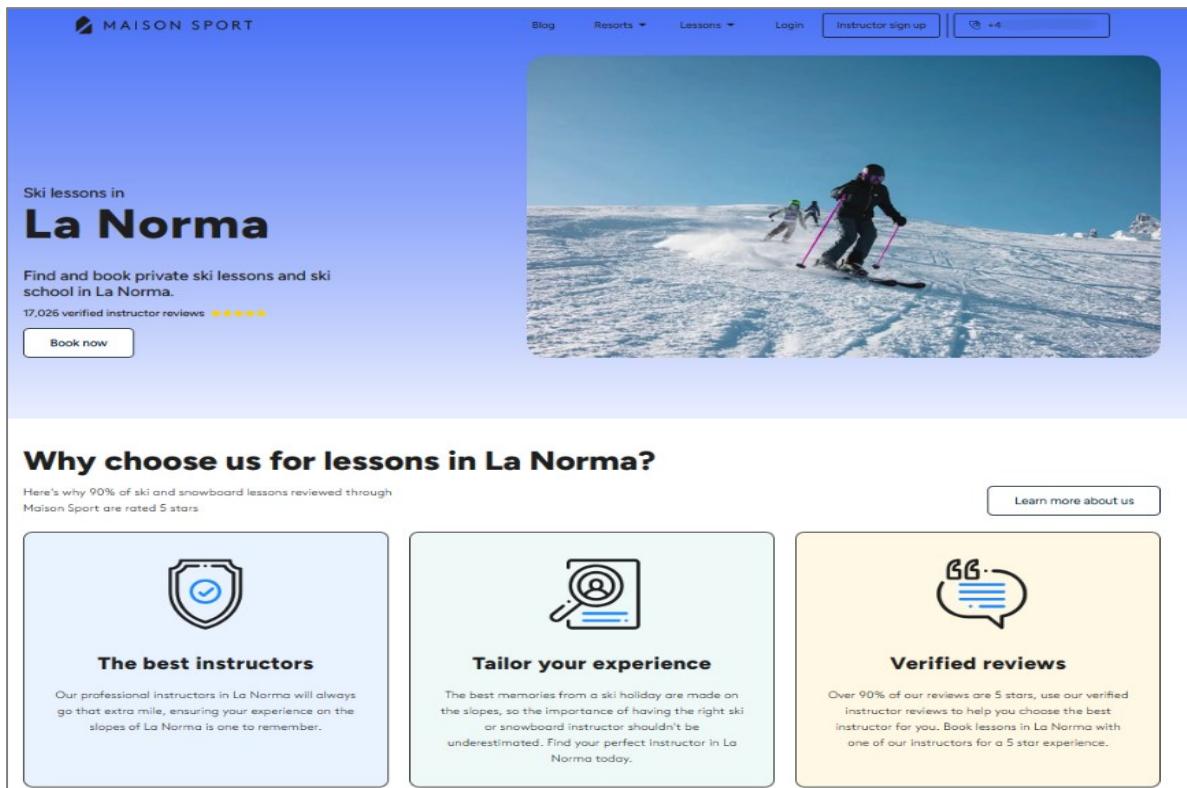


Image 7. The Maison Sport app.

Masonsport<sup>1</sup> is probably the most popular application in Europe for winter sports course planning. It includes the ski resorts of 10 European countries and focuses, among others, on France, Italy and Switzerland, i.e. in the Alpine region. The user chooses whether he is a coach or a student when registering. Then, if he is a student, he can choose the ski resort he is interested in, the desired dates and type of sport, such as off-piste skiing or visually impaired skiing. These functions are specified by the form shown in the following image.

<sup>1</sup> <https://maisonsport.com/en/resort>



## Unsure on how to make a booking?

Don't worry, our customer success team is here to help. You can contact us via phone, live chat or WhatsApp.

[Phone](#) [Live Chat](#) [WhatsApp](#)

## Let's get started

Enter your resort, dates and sport to view all available lessons.

Select resort

Trip dates

Ski touring

12.329 verified instructor reviews ★★★★★

Reserve now with a 20% deposit [\(i\)](#)

[Next](#)

Image 8. Course parameter selection form.

In fact, after the user sets his preferences, a list of available coaches and information about the courses, prices and details of the coaches is displayed, as shown in figure 9. In addition, the default option is to show the coaches with the best ratings first. Therefore, this application has several useful features. However, there are not always lessons available in all ski resorts of the countries mentioned and it is not available in Greece.

Pick your dates      Choose your instructor      Select lessons

### Quick filters

15 available instructor(s) that match your search and 2 available group lesson(s).

Private    Group    Female    Male    Morning    Afternoon    Full day    Days    More filters    Price: High to Low

**Anja M.** 5 4.99 (221 reviews)

25 years experience  
I speak: English

My lessons are divided on 3 hours in the morning or afternoon for individuals or groups. I also offer full day lessons. For small groups, price stay the same.

From €600/day  
From €100/hour

[View lessons](#)  
[Profile](#)

**Mike G.** 5 4.99 (90 reviews)

18 years experience  
I speak: English, French

I provide ski lessons whatever your level and your expectations... My motto: Learn while having fun!

From €526/day  
From €80/hour

[View lessons](#)  
[Profile](#)

Image 9. Coaches available for customer's preferences.

## 2.2.2 CheckYeti App

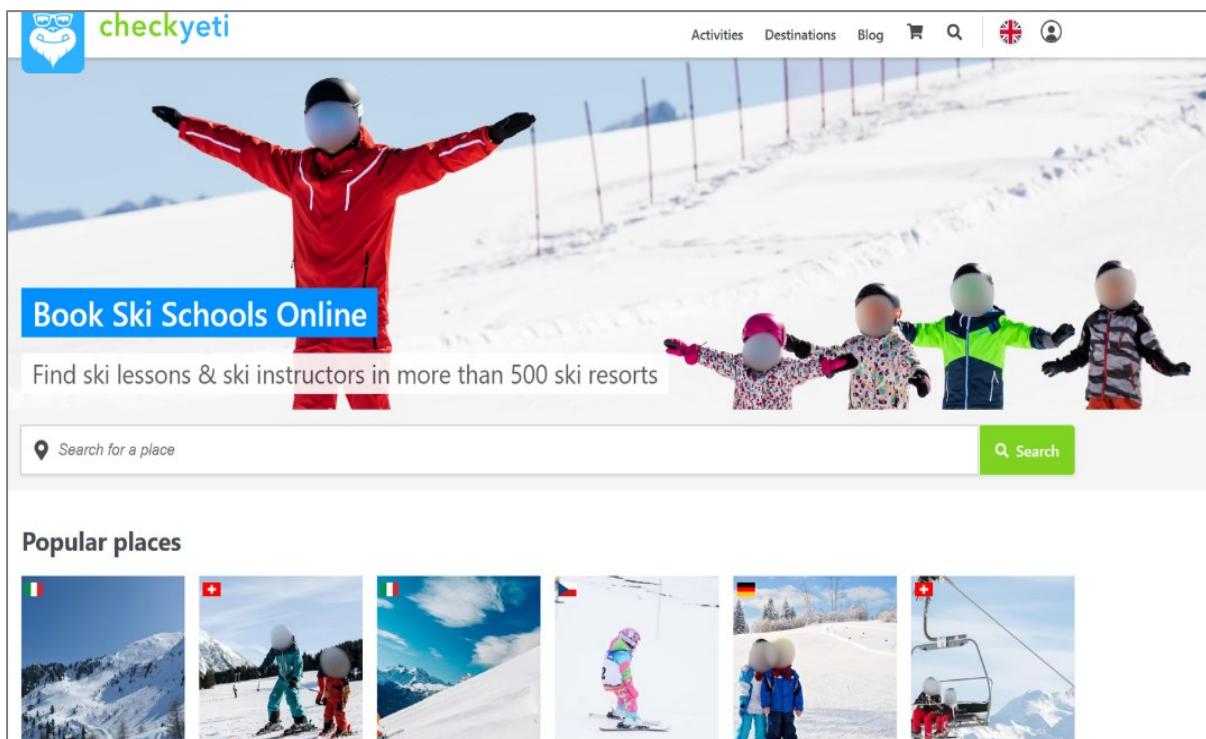


Image 10. checkYeti app.

The CheckYeti app<sup>1</sup> is an app that provides support for various sports such as diving, windsurfing, skiing, etc. The interface for scheduling winter lessons is shown in figure 10. The user can select the ski resort he is interested in and if the app supports it, it shows him recommended courses, as shown in figure 11. However, the user is not given the opportunity to find out who the coaches are in every lesson, and neither can he choose the coach he prefers.

<sup>1</sup> <https://www.checkyeti.com/en/v/1/winter>

The screenshot shows the checkYeti app interface for 'Ski Schools Austria'. At the top, there's a flag icon for Austria and the text 'Ski Schools Austria' followed by '1000+ Offers for Ski Lessons with the Best Prices 2024/2025'. A dropdown menu says 'Sort by: Most popular'. On the left, there's a sidebar with 'Select course date' (with a 'Start date' dropdown), 'Popular Options' (checkbox for 'Free Cancellation'), and 'Locations' (checkboxes for Tyrol, Salzburg, Skicircus, Ski Arlberg, and Clavia). The main area displays two ski lesson offers:

- Kids Ski Lessons (5-14 y.) for First Timers** at 'Ski school Lechner Zell am Ziller'. It shows a photo of a child learning with an instructor, a 'Just booked' button, 147 recommendations, and a green 'Check availability' button. It also mentions 'Ski hire available' and a discount 'Save up to 5% From 52€ 49€ / day (2hrs)'.
- Kids Ski Lessons (4-14 y.) for First Timers** at 'Skischule Stubai Tirol'. It shows a photo of children learning, a 'Just booked' button, 324 recommendations, and a green 'Check availability' button. It also mentions 'Ski hire available' and a price '48€ / day (4hrs)'.

Image 11. Ability to schedule lessons in the checkYeti app.

### 2.2.3 Freedom Snowsports App

The screenshot shows the Freedom Snowsports website. At the top, there's a 'Book Now' button with a shopping cart icon, and contact links for 'Call Us', 'Email', and 'WhatsApp'. The main heading is 'Ski & Snowboard Lessons in Chamonix'. Below it, there are six course planning options arranged in a grid:

- Private Ski & Snowboard Lessons**: Shows a person in a blue jacket. Text: 'The most flexibility and performance.' with a 'Book online' button. Includes a 'Private Lessons' button with an arrow.
- Adult Group Ski Lessons**: Shows a group of people on skis. Text: 'Adult coaching groups.' with a 'Book online' button. Includes an 'Adult Group Ski Lessons' button with an arrow.
- Children's Group Ski Lessons**: Shows a child on skis. Text: 'Scheduled on all holiday weeks.' with a 'Book online' button. Includes a 'Kid's Group Ski Lessons' button with an arrow.
- Weekend Group Ski Lessons**: Shows a group of people in a snowy setting. Text: 'Convenient weekend groups.' with a 'Book online' button. Includes a 'Weekend Groups' button with an arrow.
- Off Piste Skiing & Snowboarding**: Shows a person in a blue helmet. Text: 'Off piste skills lessons.' with a 'Book online' button. Includes an 'Off Piste Skiing' button with an arrow.
- Backcountry Tours**: Shows a close-up of hands holding a map. Text: 'Discover the hidden gems around us.' with a 'Book online' button. Includes a 'Backcountry Tours' button with an arrow.

Image 12. Course planning at the Freedom Snowsports school.

The Freedom Snowsports [school](#)<sup>1</sup> offers courses in a variety of ski resorts in the Alps. In fact, it provides the user with the opportunity to choose a course according to the desired sport, age, skill level and time preferences. Once the user has chosen the ski resort he is interested in, he has

<sup>1</sup> <https://www.book.ski/team>

the option of choosing a coach, as shown in Figure 13. It is a simple graphical interface that gives the user access to some information about coaches and the languages that everyone knows. Also, the available coaches are those who belong to this particular school.

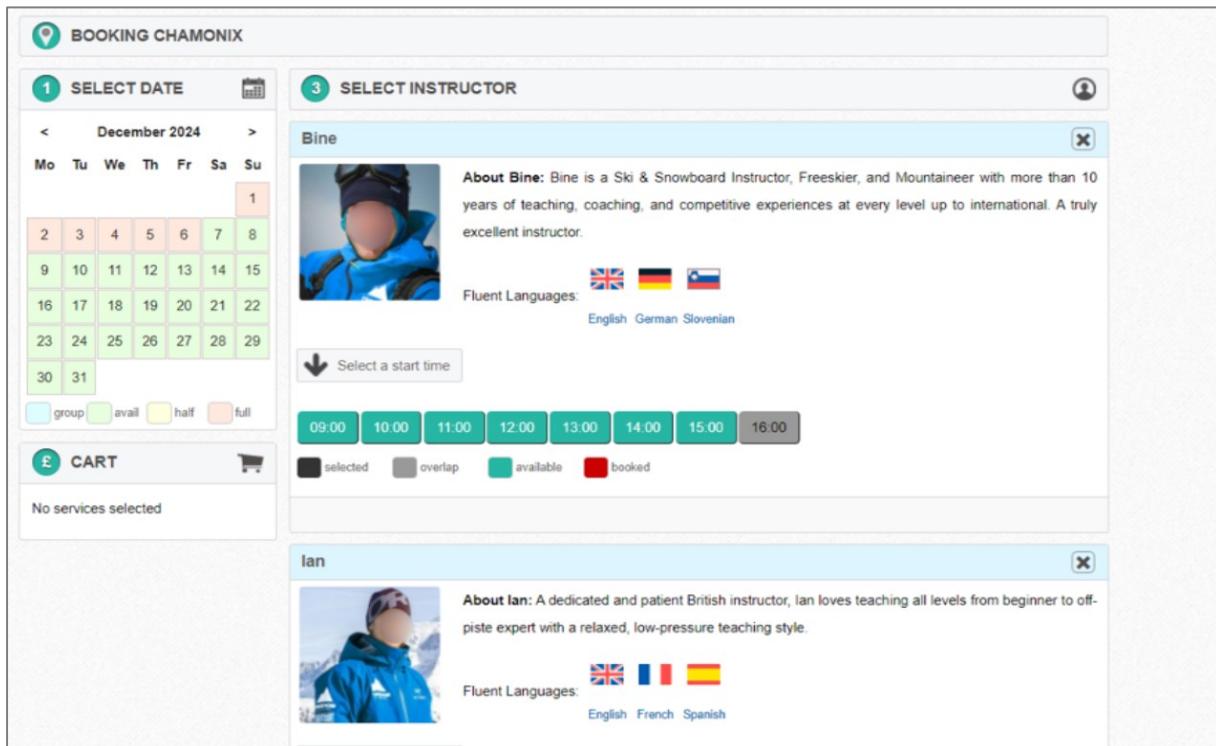


Image 13. Possibility to book a course at the Freedom Snowsports school.

## 2.3 Human-centered design

In this section, reference is made to the theory of anthropocentric design, which is the basis of the following chapters. In fact, anthropocentric design is based on some fundamental principles (Norman, 2018). In particular, the design of a system must focus on the people who will use it, meet their needs, their desires and help them achieve their goals. These are rules that are very important to apply in all phases of an application's development, and non-adherence to them is one of the main reasons why almost 70% of all software programs developed today are considered not to have achieved their original goals (Powney, 2024). For this reason, we will analyze and use the principles of human-centric design and explain the concept of usability and user experience.

### 2.3.1 Usability

Usability is defined as the ability of a product used by defined users, with defined objectives, under defined conditions of use, to be effective, efficient and provide subjective satisfaction to its users (International Organization for Standardization, 2019). This is a very important concept as it shows us that we must first of all study the users of the application, understand the needs they have and the reasons for using the application. Also, the definition shows that it is important to evaluate the system in order to study whether it meets the requirements defined, i.e. whether it

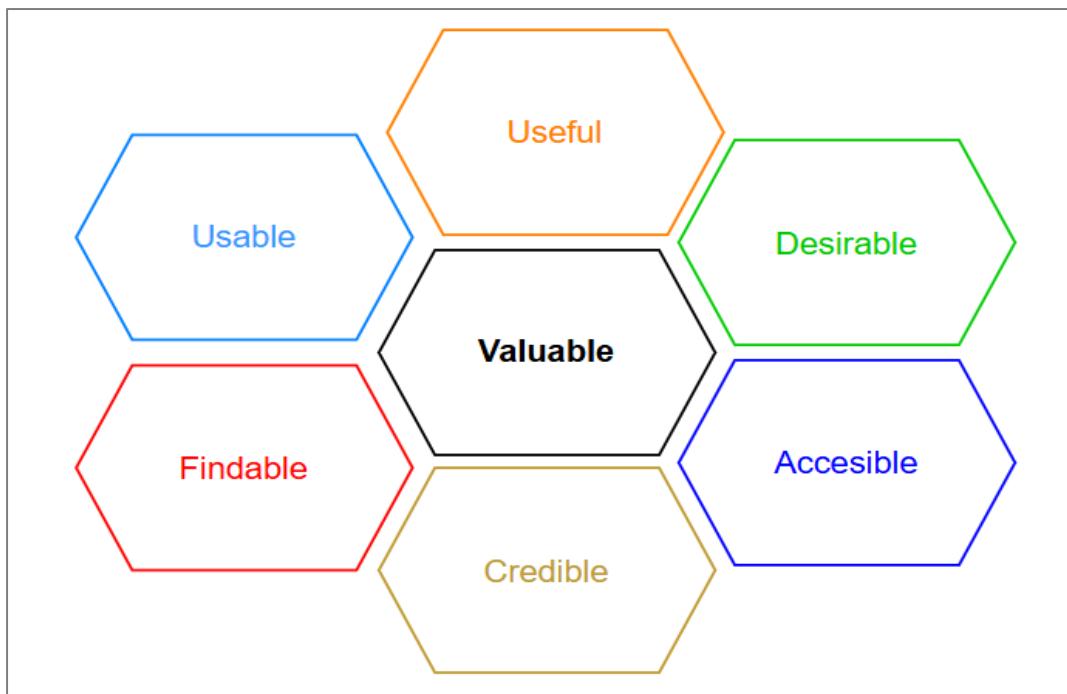
is effective, and to study whether it has unnecessary features that make its use tedious and time-consuming, i.e. whether or not it is efficient. Finally, it must also be studied whether the system offers subjective pleasure to its users.

Usability is important and for this reason other definitions have been given. In particular, usability can be considered to consist of five elements (Nielsen, 1993). The first element is the ease of learning the system, and the second is to allow the user to be productive after learning the system. The third element is that the user can easily remember the system, that is, if he tries to use the system again after some time he can do it easily without having to learn it from scratch. The fourth element concerns error management. In particular, the system must prevent users from making mistakes and in case of an error they can easily correct it. The last characteristic is the subjective satisfaction that the system offers to the user. In conclusion, both definitions of usability focus largely on the same concepts, but we will prefer the first definition.

### 2.3.2 User Experience

User experience is a term coined by researcher Don Norman in 1993 that expands the definition of usability (Nielsen, 2017). Specifically, user experience is about the value the product offers to the user himself, the emotions it creates for him and how much it makes the user want to use it.

In fact, there is a model called a usability cell that visualizes the seven characteristics of the user experience (Morville, 2016), as shown in Figure 14. According to this model, the system must above all be useful to the user and use it to achieve something. It must also be easy to use, as we defined earlier. Also, the system must be desirable, i.e. it must have a beautiful design that is visually pleasing. Of course, the system must be accessible, i.e. meet the needs of people who have, for example, vision, hearing or mobility problems. Also, depending on the functions of the app, it may also need to be accessible from screens of different sizes, such as mobile phones or tablets. Of course, the system must be credible, i.e. trustworthy, provide security as well as protect users and their personal data. Furthermore, the system and its information must be easily accessible, i.e. the content must be well structured so that the user can easily find what he is looking for. Finally, it is important that the program is valuable to the user, i.e. that he feels that he wants to use it, e.g. because it creates pleasant feelings for him or because it makes it easier for him to do a task and saves him time.



*Image 14. UX Honeycomb v. (Morville,2016).*

### 2.3.3 Principles of human-centered design

The ISO 9241-210 standard defines six fundamental principles of anthropocentric design, which we will follow in our design (International Organization for Standardization, 2019).

1. Design must start from the human being, i.e. from the understanding of the typical user and his tasks. This is especially important since users are different from each other and may be considered good for one person and not for another.
2. Users are involved in all phases of system design and development. In fact, users actively determine the requirements for the interface and features of the system, which is why designers must interact with users frequently.
3. The design is determined and modified based on the human-centered assessment. In particular, the evaluation must be carried out at all stages of development and not only at its end.
4. The development and design process is iterative. This is because it is extremely difficult, even impossible at times, for users to explain what they want from a system from the start.
5. The design is about the entire user experience. This means that the system must meet all seven characteristics analyzed in the usability cell in Figure 14.
6. The design team is multidisciplinary. This means that in order for a product to succeed, there must be cooperation between people with different skills. For example, not only software developers and designers, but also psychologists or philologists can help to design a new application for communication.

#### 2.3.4 Asteroid model

For the implementation of this work, the Hix-Hartson asteroid model (Hartson & Hix, 1986) shown in Figure 15 will be applied. According to this model, the process can start from any of the phases of requirements analysis, task analysis, design and development of prototypes and implementation of the system. Of course, the key element is that each phase is completed with an evaluation. This allows for a lot of flexibility as any overlooks can be easily found and incorporated into the final product with minimal repetition of previous steps.

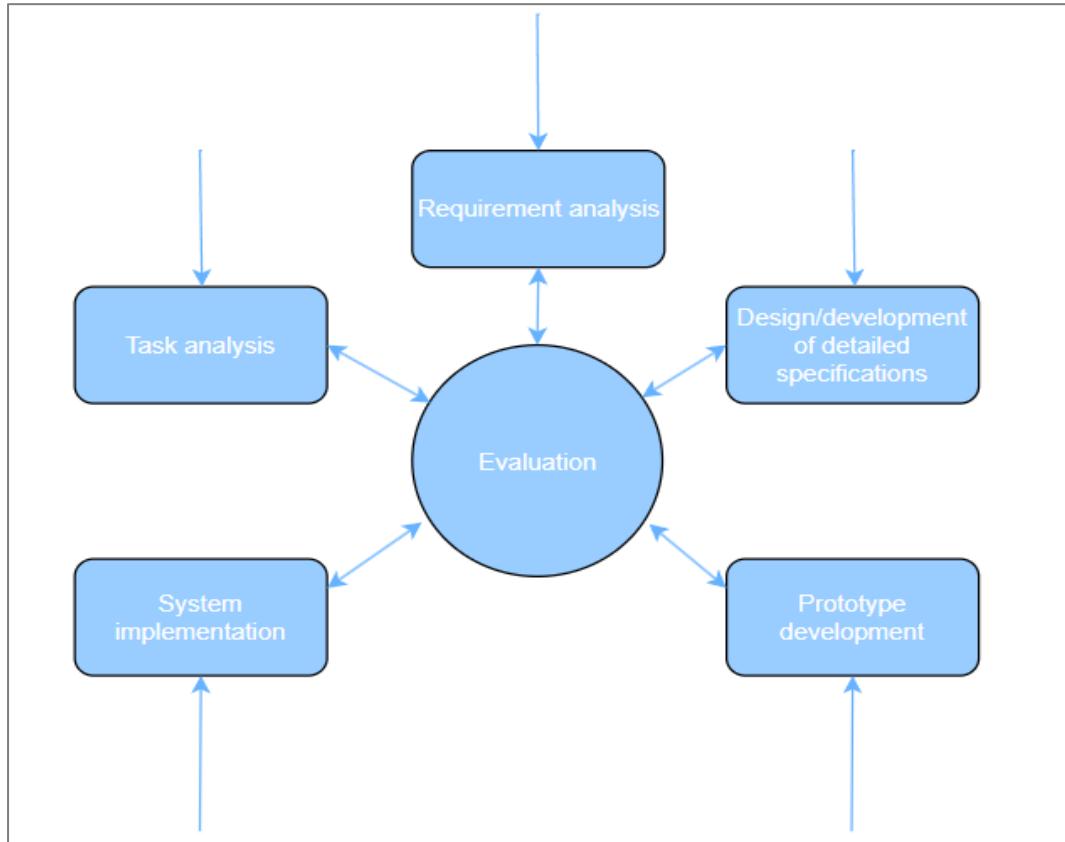


Image 15. Asteroid model (Hartson & Hix, 1986).

## 2.4 Conclusions

From the above analysis, we conclude that there are currently basic deficiencies in the organization of courses in the ski resorts of Greece. In fact, in most cases, users simply schedule their lessons over the phone with a random school, without knowing the coach they will be teaching with, their characteristics or their reviews. Other times, again, skiers visit the ski resort first and look for a coach there, which takes time and at the same time it is not certain that a coach will be found or even if it is found that he will be the best available.

Therefore, the importance of developing an application to solve the above problems is understood. The principle of this process, according to anthropocentric theory, is the analysis of the users of the application, their characteristics, the requirements they will have and the framework of use of the application, which are done in the next chapter.

Finally, in this chapter, the basic principles of anthropocentric design were analyzed and the asteroid model was presented, on which the rest of the work will be based.



### 3 Methodology

This chapter is a summary of the methodology, which is shown in Figure 16, which will be applied during the implementation of the diploma.

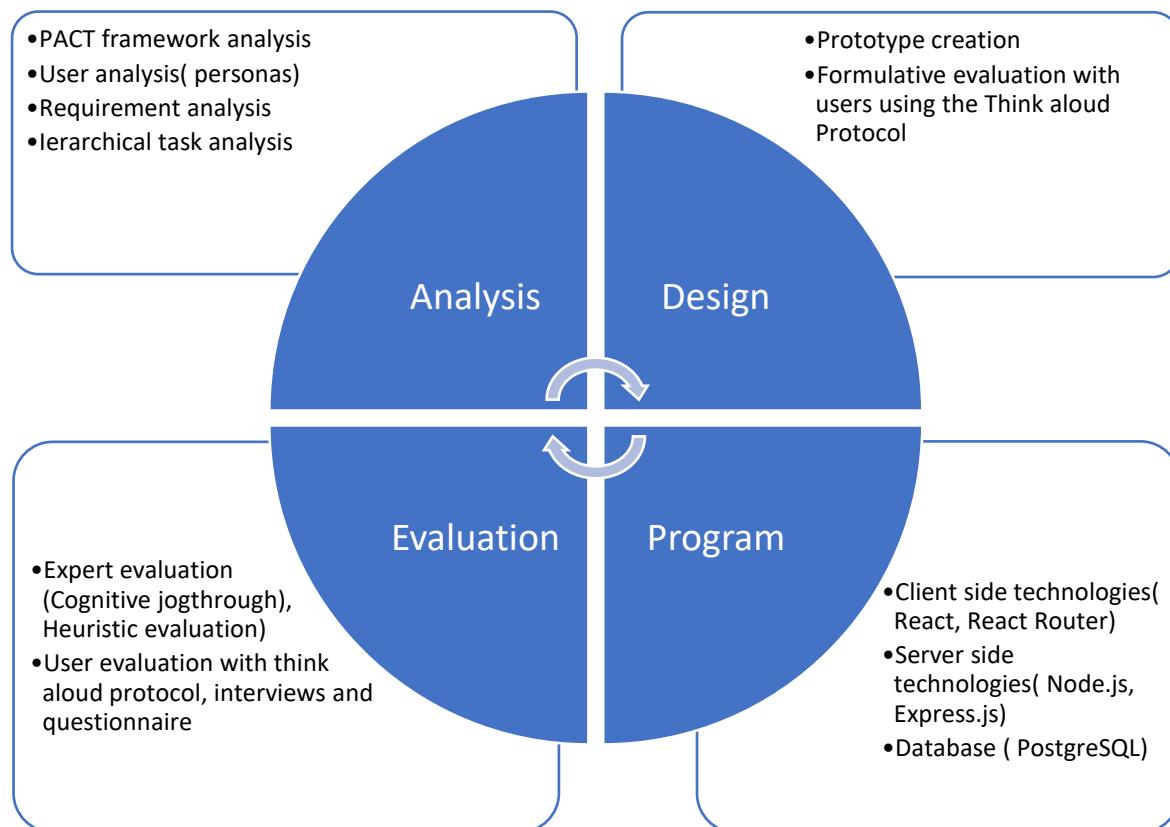


Image 16. Methodology that will be applied in the implementation of the diplomacy.

#### 3.1 Problem Analysis

Initially, an analytical approach to the problem is carried out, with the aim of defining the basic parameters and fully understanding the framework in which the application will be developed. For this purpose, the analysis of the PACT (People, Activities, Contexts, Technologies) framework is chosen as a methodological approach, which allows the systematic study of the interactions between the users of the application, their activities, the user environment and technologies (Benyon, 2019).

In fact, in this context, special emphasis is placed on understanding the typical users of the application, through the persona technique. This technique allows the creation of representative user profiles, which help to identify the needs, expectations and behaviors of users when interacting with the application.

At the same time, a requirements analysis will be carried out in order to clearly identify the desired functions and features of the application, both in terms of functionality and user experience.

Finally, a hierarchical analysis will be applied to the basic functions of the system, in order to analyze the processes in smaller and manageable steps. Through this approach, the individual actions that users are asked to perform will be understood in depth, which will facilitate both the design and implementation of the application.

## 3.2 App design

This is followed by the design of the application, in which detailed and high-fidelity prototypes are created. The creation of such prototypes is a particularly important stage, as it offers the opportunity to examine different design approaches and choose the most suitable one, through a creative and iterative design cycle. After all, prototypes function as 'consumable' designs, allowing flexibility to modify them without requiring immediate implementation in code.

At the same time, the existence of prototypes facilitates the early evaluation of the design by typical users, offering the possibility of identifying potential problems already in the early stages of development. That is, through this formative evaluation process, it is possible to intervene and correct any weaknesses in a timely manner, contributing to the improvement of the final user experience.

The Talking Subject Protocol will be used as the basic method of evaluating prototypes, whereby users express their thoughts, feelings and impressions aloud while navigating the app's designs. This methodology allows the understanding of user interaction with the application and is extensively analyzed in the corresponding section of the thesis.

## 3.3 Code implementation

Then, the implementation of the prototypes with code will take place, with the aim of creating a fully functional application. The deployment will cover both the client-side and server-side sides, ensuring the coherence and efficiency of the final system.

On the frontend side, modern React and React Router technologies will be utilized, offering high efficiency, flexibility and the ability to create interactive user interfaces. In fact, React allows the development of reusable components, while React Router facilitates the management of routes and navigation within the application.

Also, on the backend side, Node.js and Express.js technologies will be used, which are a powerful and proven solution for the development of web services. After all, Node.js allows the server to be deployed in a JavaScript environment and Express.js simplifies the creation of RESTful APIs and the management of HTTP requests.

Finally, to store and manage the application's data, the PostgreSQL database will be used, which provides high reliability, flexibility in data modeling, and advanced features such as support for JSON and indexes.

It is worth noting that all the above technological options are analyzed and documented in detail in the corresponding chapter, in order to justify their suitability in terms of the requirements and particularities of the application.

### 3.4 Application Evaluation

Finally, an evaluation of the application will be carried out, with the aim of ensuring the quality, usability and overall user experience. In fact, a combination of assessment methods by both experts and end-users will be used in order to cover multiple aspects of the interaction and functionality of the application.

On the expert side, established methods will be applied, such as Heuristic Evaluation, which is based on a set of usability rules, as well as Simplified Cognitive Routing, which focuses on how a hypothetical user interacts with the system.

At the same time, special emphasis will be placed on evaluation by end-users. In particular, the Talking Subject Protocol will be used, in which users express their thoughts while navigating the application, as well as the method of semi-structured interviews, which allows the understanding of the user experience while using the application. In addition, a questionnaire will be used, which will collect quantitative data on the usability and design of the application.

Of course, in case problems are identified in the implementation during the evaluation, these will be analyzed and corrected accordingly. Finally, the necessary improvements will be incorporated into the final version of the app, with the aim of ensuring the best user experience.



# 4 Analysis

As we mentioned in chapter 1, the purpose of this paper is to implement an application that will improve the experience of students and coaches who plan lessons in ski resorts in Greece. Furthermore, in Chapter 2 it was understood that this process today has various shortcomings. Therefore, it is extremely important that this application is properly designed to significantly improve the experience of students and coaches. To make this possible, in this chapter we will analyze the implementation parameters and initially define the theoretical background for the PACT framework. Then we will do user analysis and task analysis. User analytics allows us to understand who the users of our app are and what particular features they have. In fact, for the analysis of the users we will define the stakeholders for the application and we will use the persona technique. Also, task analysis aims to understand what users expect from the app and how they will use it to achieve their goals. In fact, in order to analyze tasks, we will first study the context of use of our application. Next, we will analyze the actions that users take with the technique of hierarchical analysis. We'll also define the specific features that our app needs to support. Finally, we will use the results of the analysis to determine the PACT framework of the application.

## 4.1 PACT Framework - Theory

The PACT framework is a key element in human-centered design that allows us to understand the users of the app, the actions they want to take, the context in which they will use the app, and the technology at their disposal (Benyon, 2019). In fact, it is based on the following sections:

**People:** They are the users of the system. Understanding them is very important because people are not all the same and there are physical, psychological, mental, social and value differences between them. Therefore, different users may consider different things important or some may like a design that some others don't like.

**Activities:** This refers to the activities and actions that users will take. Understanding users' activities is important, as they may vary in frequency of execution, the existence of time or other limitations, the way they are performed (consecutive or in parts), their nature (individual or group) and the potential risk of an accident in the event of an error.

**Context:** This is the context of use of the application. It refers to the space in which users are located, whether they are alone or with other people, whether they use the application in their free time or at work, etc.

**Technologies:** This refers to the interactive technologies that the user will come into contact using the application. For example, the user can use an application in a special kiosk, on his mobile phone or on a desktop computer.

The parameters of the PACT framework for our implementation will then be defined.

## 4.2 User Analysis

The first step in human-centric design is understanding typical users and their needs. For this reason, the stakeholders are initially defined and then personas and representative usage scenarios are created.

### 4.2.1 Stakeholders

Central to the definition we gave for usability in section 2.3.1 is the users of the application. Therefore, in this section we will define the stakeholders, i.e. all the people affected by the application that will result from the design process. In fact, the interested parties are divided into three categories of users, the primary, the secondary and the tertiary (Avouris et al. 2018):

- Primary users are those who are going to interact frequently and directly with the system.
- Secondary users are those who use the system less frequently or through an intermediary.
- Third-party users are people who never use the system directly, but are affected by its introduction, since they are obliged to change some of their tasks in order to adapt to its requirements.

Therefore, for our app, Easy Snow, users are divided as follows:

**Primary users** are winter sports coaches and skiers who use the app frequently.

**Secondary users** are the app's development and maintenance team, which consists of the author, and skiers who may create an account but not schedule a lesson or rarely use the app.

**Tertiary users** are those who are affected by the creation of our application. Therefore, all winter sports schools in Greece belong to this category. Also, tertiary users are the people who are involved in winter equipment rental, since the application can push more people to try and engage in a winter sport in order to increase the need for equipment. for exactly the same reason, the Management and Operation team of the Ski Centers is included in this category of users, as they will have an increase in the number of visits.

### 4.2.2 Personas

To understand the users of the application we will use the persona technique (Cooper, 2004). Personas are personas that cater to specific categories of typical users and allow the designer to put themselves in the shoes of future users, see the app through their own perspective and really understand what needs and expectations users have. This allows us to understand the specific characteristics of future users and to realize the goals of each category of users and to design the application to meet them.

In fact, the personas have emerged from the author's personal experience and contact with several skiers and coaches during several years of involvement in skiing. Also, some personas

refer to typical characteristics of people recorded in the reviews on the [Maison Sport<sup>1</sup>](#) app, mentioned in section 2.2.1.

### Persona 1: First Ski Lesson

Maria is a student who wants to try a winter sport for the first time. In fact, this persona is a typical user case that has emerged from the author's personal experience and from contact with many people in ski resorts that have similar characteristics.



Image 17. Persona 2: Mary Papadatou.

### Use Scenario 1

Maria is a 22-year-old student studying Business Administration in Athens. This Christmas, she wants to go on vacation with her best friends, Danae and Eleni. In fact, they have heard from many of their friends that Arachova is very beautiful and have decided to visit them for a week. Maria and her friends are athletic and like to try new things, so they agreed to ski or snowboard. They have already procured the necessary equipment such as gloves, goggles and overalls and want to schedule lessons with a coach for all the days of their vacation. They even want to do the planning several weeks before their vacation so that they don't have to worry about finding an available coach at the ski resort. However, those coaches she spoke to on the phone, they replied that they only schedule classes a few days in advance because they can't score for more days.

### Persona 2: Lesson for the whole family

<sup>1</sup> <https://maisonsport.com/en/profile/1010946121/alessio-r>

Dimitris is a 45-year-old family man who wants his children to try skiing lessons on his family vacation. In fact, this persona is encountered quite often and has emerged from the observation of reviews on the Maison Sport app.

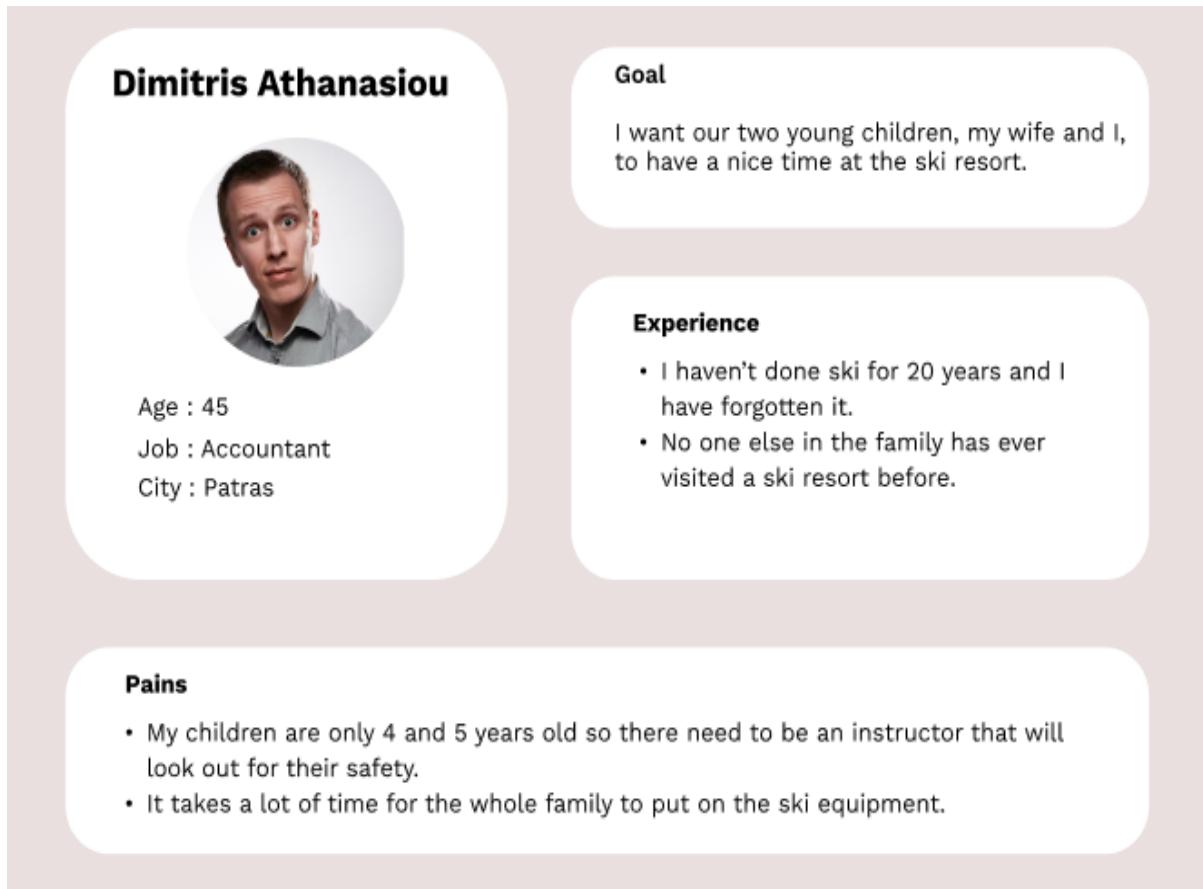


Image 18. Persona 3: Dimitris Athanasiou.

## Use Scenario 2

Dimitris is 45 years old, he is married to Anna and they have two small children, George and Thanasis. This Christmas, they will go on a family vacation to Kalavryta to see snow up close for the first time. In fact, Dimitris wants the children to come into contact with the sport of skiing in order to do something active during their holidays that they will remember. Dimitris had tried skiing once when he was a student, but it had made it difficult for him and he has never been involved again. That's why he wants to schedule a lesson with a coach who will teach his children the basics. Of course, Dimitris and his wife Anna feel ready to try skiing too, so they will all do a family lesson together. For this reason, Dimitris is looking for a ski instructor who has experience in children and makes teaching enjoyable and easy. But he finds it difficult to choose a coach because he doesn't have someone he knows who can ski and be able to introduce him to someone.

### Persona 3: Coach with a lot of experience

Michalis is a coach who has a lot of experience and delivers daily skiing and snowboarding lessons. Furthermore, this persona has emerged from the author's personal experience and

through various interactions with coaches in ski resorts. Furthermore, from the above interaction it has emerged that coaches want to avoid dealing with lesson planning when they are at the ski resort and have a lesson.

**Michael Zerbias**



Age : 34  
Job : Ski & Snowboard instructor  
City : Athens

**Goal**  
Offer high quality lessons so that everyone has a good time.

**Experience**

- I teach for over 10 years ski and snowboard at Kalavryta ski resort.
- I also teach competitive groups.
- I know all slopes and all dangerous sports of the mountain.

**Pains**

- I have no way to inform those looking for a coach about my experience or the very positive feedback I receive in my lessons.
- I don't want to answer phone calls for scheduling lessons while I'm in a lesson.

Image 19. Persona 4: Michael Zerbias.

### Use Scenario 3

Michalis is a ski and snowboard coach with 12 years of experience. He chose his profession because since he was a child he liked to go down the various tracks. In fact, he has always wanted to try new things and has gained experience in off-piste descents, untouched snow and bad weather conditions. Of course, the main reason he decided to pursue teaching skiing and snowboarding professionally after graduating from the T.E.F.A.A school of Trikala is because he likes to get in touch with people and teach them how to enjoy the mountain. Indeed, his students, whether they are young or older, have repeatedly commented on him how pleasant, calm and fun his lessons are. In fact, many students have recommended Michalis to their friends so that they too have the opportunity to gain a pleasant experience from the mountain. However, Michalis wants to be focused during class time and that's why he doesn't answer the phone when he's doing a lesson to those who call him to schedule a lesson. However, when Michalis calls them after class, some of them have already scheduled a lesson with another coach.

### Persona 4: Coach with knowledge of 3 foreign languages

Artemis is a ski and snowboard coach who knows three foreign languages quite well, namely English, French and Italian, so she can easily communicate with tourists. In fact, this persona has emerged from the projection of the characteristics of the coaches in the Maison Sport app.

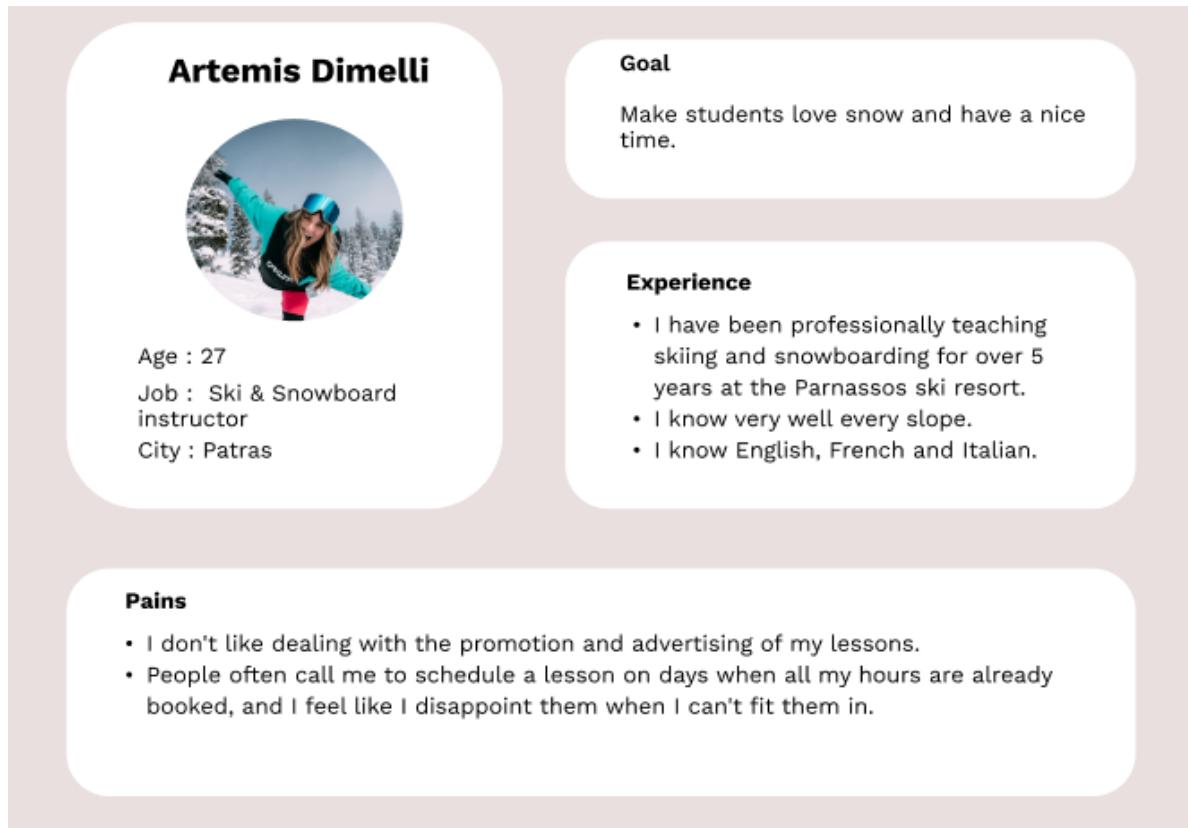


Image 20. Persona 5: Artemis Dimelli.

#### Use Scenario 4

Artemis is 27 years old and has been teaching ski and snowboard lessons for 5 years at the Parnassos Ski Center. He considers the Parnassos ski resort to be one of the most beautiful in Greece and this is the reason why so many tourists from abroad visit it. In fact, most tourists know English but some only speak their native language, which is mostly French or Italian. Artemis, of course, can easily communicate with most tourists as she speaks 3 languages: English, French and Italian. In fact, there are few coaches who can communicate as well as Artemis and that's why Artemis' colleagues, when they don't know the mother tongue of tourists, refer them to her. After all, Artemis is very friendly and sociable and loves to take lessons regardless of whether her students know Greek. Of course, Artemis is not familiar with the use of the internet for the scheduling of her lessons and she does not want to be involved in the advertising of her lessons. Also, Artemis receives frequent phone calls about scheduling classes on days when all her hours are closed and feels that she disappoints her students when she informs them that she does not have time to schedule a lesson together.

### 4.3 Requirements Analysis

Once the standard users and usage scenarios of the app have been defined, the requirements that the app must meet can be defined. Requirements are divided into functional and non-functional requirements. In fact, functional requirements determine what the application should do, and non-functional requirements describe how to do it (altexsoft, 2023). In particular, the functional requirements arise from the goals of the users and the reasons they use our application. Non-functional requirements result from system usability and user experience, which were analyzed in section 2.3.

Table 3 lists the basic functional requirements of the Easy Snow application in order of priority and the estimated duration of their implementation.

*Table 3. Table of functional requirements.*

n/a	Functional Requirement Description	Priority	Estimated duration
1	Create an account	High	7 days
2	Creating Courses by Coaches	High	7 days
3	View available coaches by criteria (location, dates, sport, etc.)	High	5 days
4	Scheduling Lessons with Coaches	High	5 days
5	Create user profiles	High	5 days
6	Change profile data	High	2 days
7	Overview of scheduled user courses	High	5 days
8	View Coaches Program	High	5 days
9	Ability to cancel classes	Moderate	5 days
10	Overview of Coach Experience and Skills	Moderate	5 days
11	Overview of Coaches Reviews	Moderate	5 days
12	Create a Coach Review	Moderate	2 days
13	View course history	Moderate	5 days
14	Ability to specify application language	Moderate	7 days
15	Ability to send queries to coaches	Moderate	3 days

n/a	Functional Requirement Description	Priority	Estimated duration
16	Ability to set a course cancellation policy	Moderate	3 days
17	View statistics from coaches	Low	5 days

The functional requirements mentioned in Table 3 are then analyzed.

### **Requirement 1**

The first requirement is for users to create an account so that they can interact with the app, store their information and be able to access it from any computer, mobile or tablet.

### **Requirement 2**

The second requirement concerns the creation of courses by coaches that can be a single lesson e.g. a Saturday or be repeated, i.e. every day for a period of time.

### **Requirement 3**

The third requirement is that skiers be able to see the available coaches according to their preferences and possible limitations. That is, they should be able to choose the ski resort, sport, participants and desired dates and see the available coaches.

### **Requirement 4**

The fourth requirement is that skiers should be able to schedule a lesson with the coaches.

### **Requirement 5**

The fifth requirement concerns the ability to create user profiles, for example to specify their name, profile picture, email address and mobile phone.

### **Requirement 6**

The sixth requirement is the ability to modify the users' profile. For example, a user may change a mobile phone or have entered an item incorrectly when registering and wish to change it.

### **Requirement 7**

The seventh requirement is the ability for users to be able to see the future lessons they have scheduled.

### **Requirement 8**

The eighth requirement concerns coaches, who should be able to see their schedule for a specific day.

### **Requirement 9**

In addition, the ninth requirement is course cancellation. In particular, coaches may wish to cancel a lesson due to a sudden obligation. Accordingly, skiers will be able to cancel a lesson depending on the cancellation policy chosen by each coach.

### **Requirement 10**

The tenth requirement is that students can see for each coach the skills and experience they have.

### **Requirement 11**

Requirement eleven is the ability to view the reviews that a coach has.

### **Requirement 12**

Requirement twelve is for users to be able to evaluate the coaches they have studied with by writing a review.

### **Requirement 13**

Requirement thirteen is for viewing course history, i.e. users should be able to see which courses they have participated in in the past.

### **Requirement 14**

Requirement fourteen applies to users who may be tourists and do not know Greek, so they should be able to choose the language of the application.

### **Requirement 15**

The fifteenth requirement is that students have the ability to send queries to coaches through a contact form.

### **Requirement 16**

Requirement sixteen is the ability to determine the policy of cancellation of coaches' courses. For example, a coach may allow the cancellation of a lesson up to one week before the lesson.

### **Requirement 17**

Requirement seventeen is for coaches to have access to statistics for their courses, e.g. how many hours they worked in the previous month.

Table 4 then lists the non-functional requirements of the application.

*Table 4. Table of non-functional requirements.*

n/a	Non-Functional Requirement Description	Priority
1	Easy and intuitive to use the app	High

2	Adaptability and responsiveness to different device sizes	High
3	Elegant design	High
4	Protect and secure users' data	High
5	The app should work on different browsers	High

---

The non-functional requirements mentioned in Table 4 are broken down below.

### **Requirement 1**

The first non-functional requirement is that the app's interface is easily understood and users can intuitively interact with it.

### **Requirement 2**

The second requirement is that the app be customizable and accessible from devices with different screen sizes. After all, this is one of the basic requirements of the usability cell of Figure 14.

### **Requirement 3**

The third requirement concerns the design of the app, which must be aesthetically pleasing.

### **Requirement 4**

The fourth requirement concerns the protection of users' data. In fact, user passwords in particular should be protected in any case, even if a malicious actor gains access to the database.

### **Requirement 5**

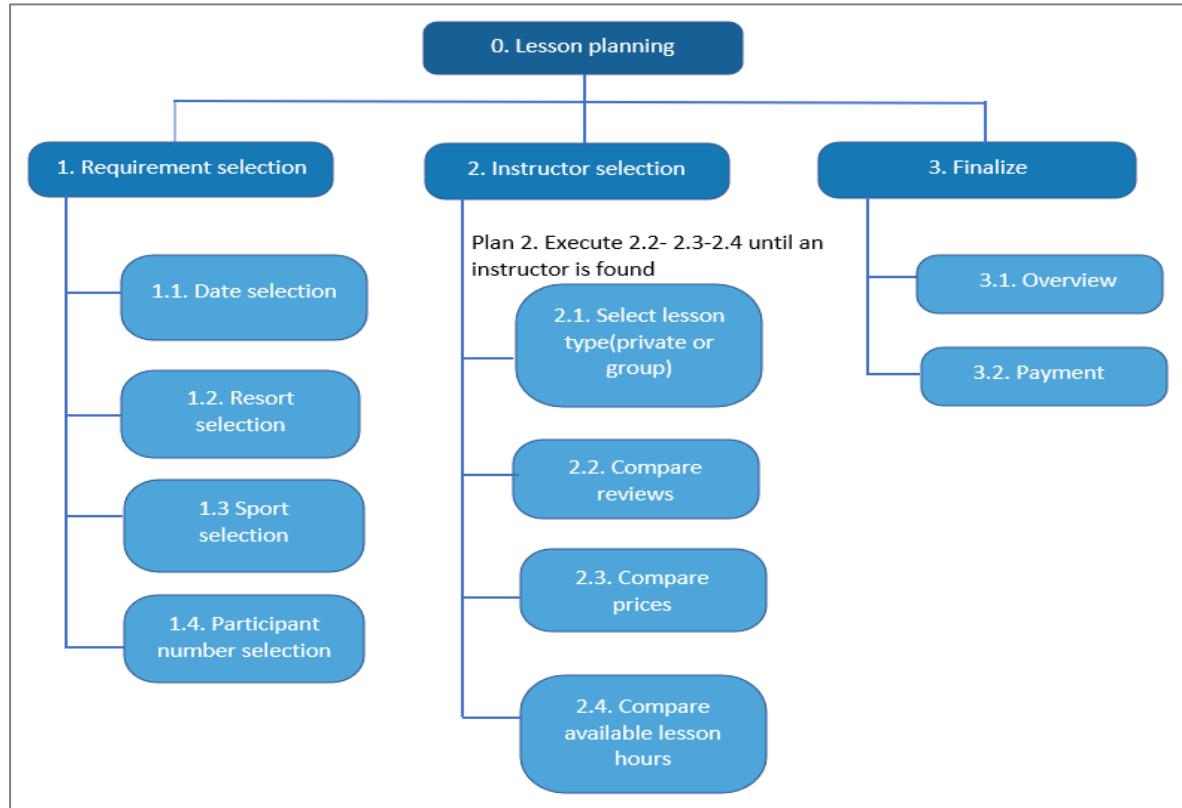
The application should be accessible from different browsers.

## **4.4 Hierarchical analysis**

Having defined the system requirements, the next step is to analyze the steps that users need to follow to realize their basic goals. For this purpose, we will use hierarchical analysis, which is an important tool for designing and studying the process that users must follow to achieve the desired result. This allows us to define in detail the steps to achieve each goal and optimize them (Hornsby, 2010). Specifically, we are studying the steps for three procedures, the scheduling of a lesson by skiers, the creation of a lesson by coaches and the supervision of the coaches' lessons for a specific day.

### **Objective 1: Lesson planning**

Skiers mainly use the app to plan lessons with coaches. The steps to follow are shown in Figure 21. First, they need to define various parameters. Specifically, they need to choose the date, the ski resort, the sport they are interested in and the number of participants. The skiers then choose their preferred coach. The choice can be based on various criteria such as the reviews that each coach has, the comparison of prices, available hours and the type of course, i.e. whether it is group or private. Finally, the skiers complete the process by reviewing the data and making a payment.

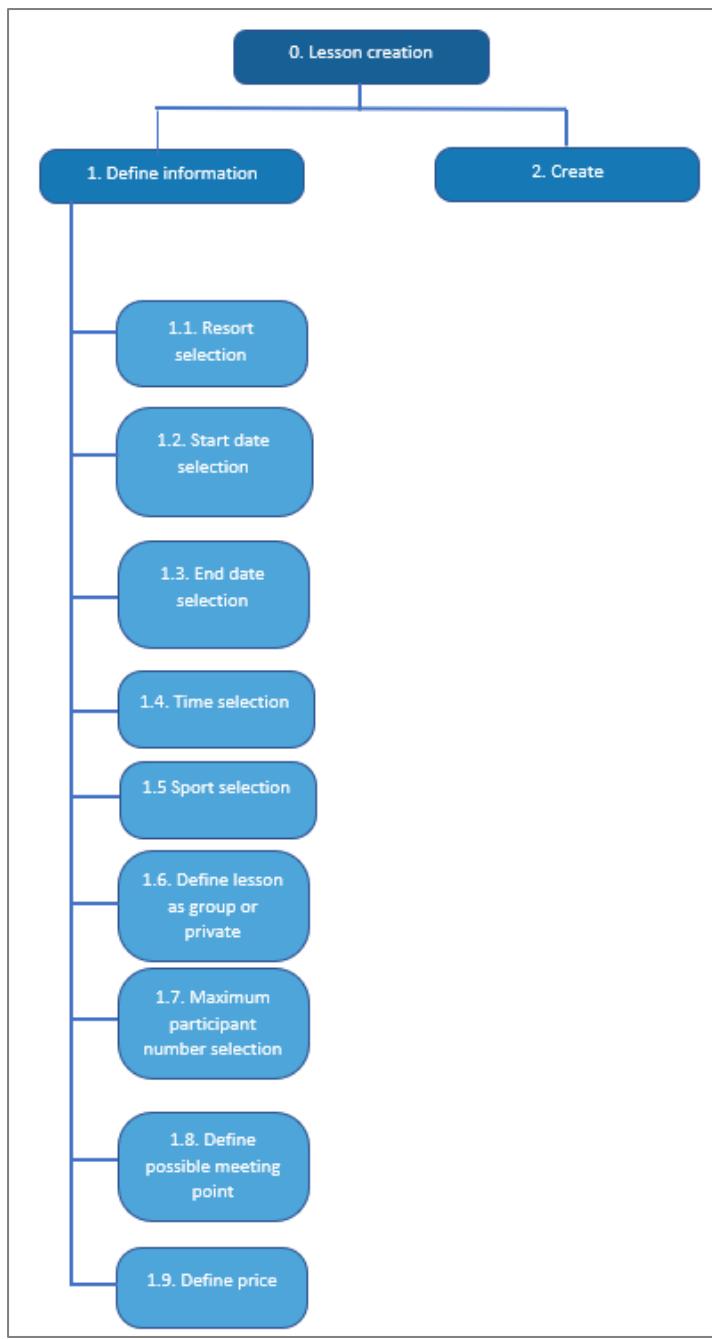


*Image 21. Hierarchical analysis of tasks for course planning.*

## Objective 2: Create a course

An important activity that coaches will do is to create lessons, and the required steps are shown in figure 22. The coaches will initially define the characteristics of the course. Therefore, first of all, they will choose the ski resort where the lesson will take place. They will then choose the start and end date for the specific course. The end date is needed in order to be able to easily create courses that are repeated, e.g. they can be done every day in a period of time. Then, the coaches will choose the time of the lesson, the type of sport and determine whether it is private or team. They will also set the maximum number of participants. They will also set the meeting point at the ski resort with the skiers. For example, a meeting point can be a chalet, a lift or the starting point of a slope. Finally, the coaches will determine the price of the course. After defining all the characteristics of the course, they will finalize the course.

In fact, the actions for this process are the final product of many iterations and evaluations, according to the asteroid model (figure 15). Specifically, in previous drafts there was no provision for choosing an end date in case of a repetitive course or for determining the meeting point.

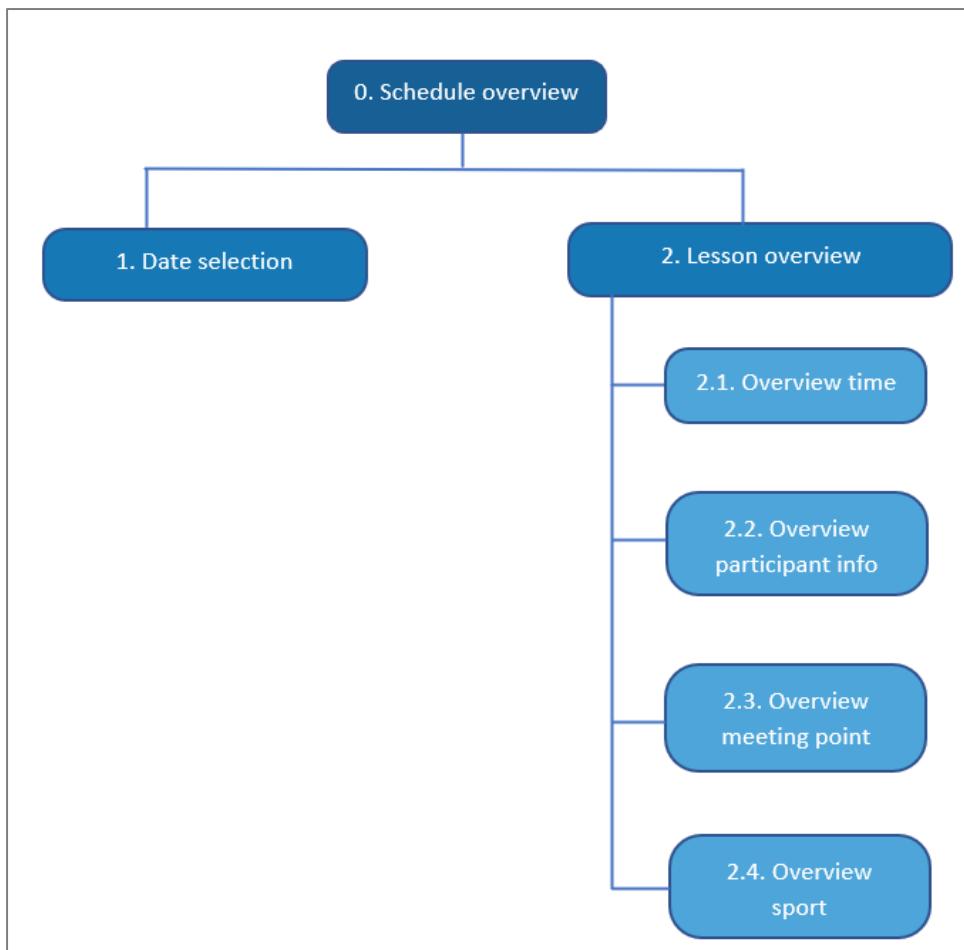


*Image 22. Hierarchical analysis of tasks for the creation of a lesson by a coach.*

### **Objective 3: Supervision of scheduled courses**

Coaches want to know their scheduled lessons. This is very important because they may want to know for the next day what their schedule is or when they are at the ski resort and finish with a lesson they may want to check what the meeting point is for the next lesson.

The steps they need to follow to supervise the scheduled courses are shown in figure 23. First, the coach chooses the date for which he wants to check the lessons. The coach can then see the desired details for each lesson, such as the time, participant details, sport type, and meeting point.



*Image 23. Hierarchical analysis for program oversight.*

## 4.5 PACT Usage Framework - Results

We then define the usage framework of our app, i.e. the environment and space in which users will use our app. Finally, we will summarize the results of this chapter by listing the individual elements of the PACT framework, which was theoretically defined at the beginning of the chapter (section 4.1).

### 4.5.1 Scope of use

The usage framework refers to the conditions in which users use our application, i.e. what space they are in, what special conditions exist and how they access the application.

Skiers interested in scheduling a lesson will usually use the Easy Snow app either from their mobile phone or from a computer to find out about available coaches depending on their limitations and preferences. In fact, we expect them to spend some time on the app to compare coaches, compare prices, and find available hours and days. Therefore, a user who will eventually schedule a lesson with a coach is likely to have visited the app multiple times, even on different days, to choose their preferred coach. Also, some users may see the available coaches without wishing to schedule a lesson directly, but because they may want to do so in the future. Of course,

skiers after planning a lesson will continue to visit the app to confirm various facts. After all, many skiers on the day they have scheduled a lesson and are at the ski resort will want to check the meeting location, for example.

Coaches will use the app for both planning their lessons and reviewing them. Therefore, teachers will connect either from their mobile phone or from the computer to the app to set the days and times when classes will take place, to be informed how many lessons are scheduled and to see various statistics. Of course, when coaches are at the ski resort, they may want to call a student or see what the meeting point they have set.

Therefore, both students and coaches can use the app from their mobile phone or a computer. In fact, it is certain that many users will use the app when they are at the ski resort, where temperatures will be around 0 °C and for this reason most users will wear gloves.

#### 4.5.2 PACT Framework - Results

At the beginning of the chapter, the PACT framework was defined and what its individual elements are, i.e. the users of the application, the functions they will do with the application, the framework in which they will use the application and the technology they will have to interact with the application. Therefore, based on the above analysis, the PACT framework for the Easy Snow application has been identified, which is listed below.

- **People:**

- Easy Snow app students are mostly people with limited or no experience who want to try a winter sport on their own or with their family.
  - Users of the Easy Snow app are also the coaches of winter sports.

- **Activities:**

- Students wish to choose coaches according to their preferences.
  - Students want to schedule lessons with ski and snowboard coaches.
  - Coaches want to create and manage their courses.
  - Both coaches and students want to get an overview of their scheduled lessons.

- **Context:**

- Students wish to organise lessons with coaches before visiting a ski resort. The visit is usually combined with an excursion to a winter destination that can last a day or more.
  - Coaches regularly use the app to organize their work and can easily supervise scheduled classes.
  - Coaches and students use the app both in a quiet environment, such as their home, and in an outdoor environment such as a ski resort.

- **Technologies:**
  - Lessons can be scheduled from a computer, tablet or mobile phone.
  - When users are at the ski resort, they will use the app from their mobile phone.

## 4.6 Conclusions

In this chapter we have analyzed the PACT framework of the application, i.e. we have defined the users of the application, the requirements they have, the context in which they will use the application and the technology they will have at their disposal. Furthermore, the steps that users should follow to carry out typical scenarios of using the application were studied.

Therefore, in the next chapter, the creation of prototypes for the application pages can begin, which will cover the functional requirements defined in Table 3.



# 5 Design

This chapter first explains the importance of design in software development and mentions some design principles that were applied. Prototypes are then designed for the application that meet the functional requirements specified in Chapter 4. Then, the designs are evaluated based on the observations of typical users, in order to identify any design problems at an early stage and correct them.

## 5.1 Importance of design

Designing and developing prototypes takes time but presents a variety of benefits that actually speed up the software development process. The benefits have been extensively analyzed (Lonec 2024; Teixeira, 2016) and the main ones are listed below. Specifically:

- Prototypes are consumable and allow for easy and quick experimentation.
- The development of designs contributes to the exploration of different design solutions.
- Prototypes are important for engaging end-users early in the evaluation process. After all, one of the principles of human-centric design, as mentioned in section 2.3.3, is that users are involved in all phases of the design and development of the system. In addition to this, the prototypes facilitate the evaluation, which is the main activity of the asteroid model presented in section 2.3.4.
- A common mistake is to directly develop interfaces using code and programming languages, resulting in identifying with the interface after we have spent enough time creating it, so we may not look to find the best design solution. This problem is addressed through the design process.
- When we want to present and promote an idea to various stakeholders to get their approval or feedback, it is much faster to do so by building prototypes than by creating the whole application in code.
- Drawings are very important in large applications, because they allow us to examine how individual elements are connected and integrated.
- When development is done by a team of developers, it is practically impossible to ensure homogeneity and consistency in the interface if there is not initially an agreed plan to develop.
- Even if the development is done by one person, it is much easier to ensure consistency in the interface if the designs have already been created than if they are specified during code development.

## 5.2 Design principles

In this section we mention some of the main design principles (Avouris et al. 2018) that we have followed.

First of all, it is important to have consistency throughout the interface, for example in the menus, icons used and in the colour codes. In fact, the arrangement of the objects must reflect the logical sequence of the user's actions. Also, the system must predict the user's actions, i.e. if there is a common value entered by users in a field, it must be set as predetermined.

Of course, it is particularly important for the application to support the user's navigation. In other words, there must be the general options in a specific location so that the user has a fixed reference point. For this reason, the app brand must be in the same position on all pages and the user must be able to tap on it to return to the homepage. After all, the navigation menus that will be developed must also be in the same position on the different pages.

It is also very important that the application has a simple and minimalist design. For this reason, a layout is created using white space and visual hierarchy. In fact, the visual hierarchy can be achieved by choosing an appropriate font color and size. Of course, in order to avoid color pollution, which tires the user, we must use only one main color in our application.

In fact, good design uses a variety of icons and images. In particular, icons are placed in lists of attributes or steps to make them easier for the user to recognize. Icons are also used for actions and possible menu options. Correspondingly, images are used to visualize a message and make the user's interaction more vivid and direct.

Following the above rules, we developed prototypes for our app, which is called Easy Snow. In fact, we chose turquoise blue as the main color. This is because turquoise refers to snow, winter and winter sports. Additionally, blue exudes stability, confidence, and calmness (Chapman, 2021).

In fact, the icons were used in the [Iconify](#)<sup>1</sup> collection, which contains a variety of icons. In addition, the app uses representations (sketches), which come from the websites [illustration kit](#)<sup>2</sup> and [unDraw](#)<sup>3</sup>. However, these websites have been chosen because of their beautiful representations as well as their open licenses. In addition, the various images come from the [pexels](#)<sup>4</sup> collections and [Pixabay](#)<sup>5</sup>, which have particularly large collections with a variety of images and open licenses.

Finally, we would like to mention that we have used, among others, the relevant applications mentioned in Chapter 2 for inspiration.

---

<sup>1</sup> <https://iconify.design>

<sup>2</sup> <https://illustrationkit.com>

<sup>3</sup> <https://undraw.co>

<sup>4</sup> <https://www.pexels.com/el-gr/>

<sup>5</sup> <https://pixabay.com>

## 5.3 Prototype design

This section presents the prototypes implemented through the [Figma](#)<sup>1</sup> design tool and comments on their functions.

### 5.3.1 Home screen

The home screen of the app is shown in Figure 24. The colors used, as mentioned, are shades of turquoise, reminiscent of snow. Also, the images are combined with the empty space to create an elegant and minimalist design. In fact, the app's brand, Easy Snow, is located in a fixed position at the top left. The user from the home screen can book with coaches or enter the app.

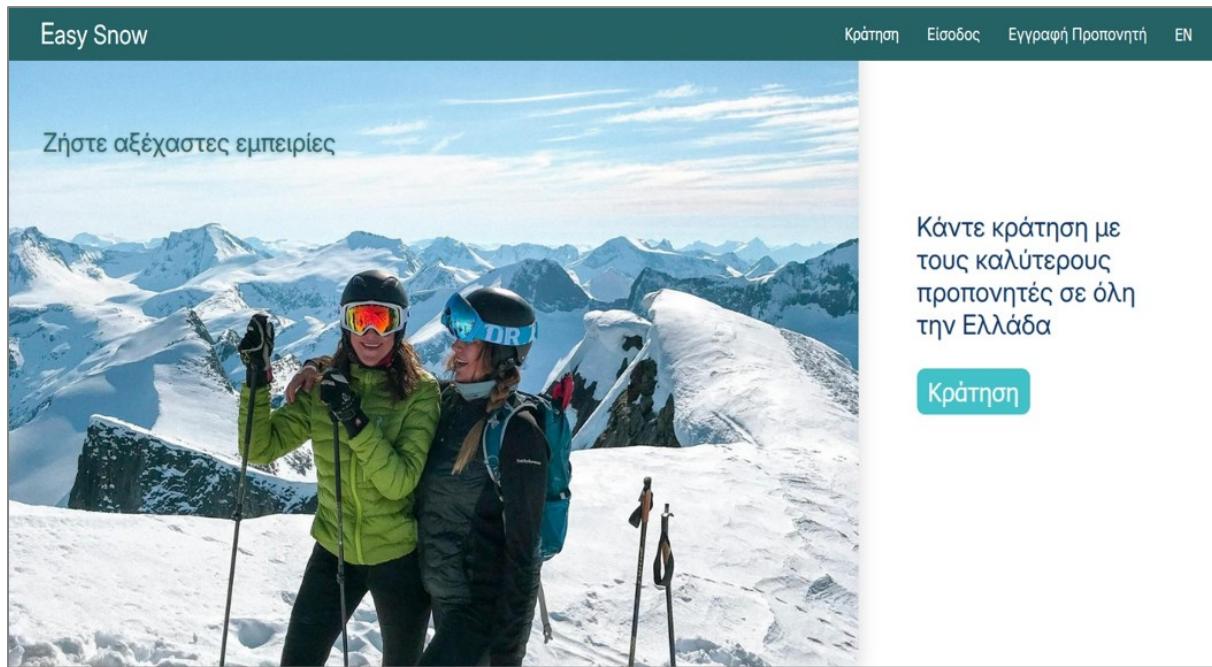


Image 24. Easy Snow app home screen.

### 5.3.2 Login to the app

By selecting Login, the user is taken to the page shown in Figure 25. Specifically, the login to the system is done with the user's email address and a password. In case the user does not have an account, he can very easily create one by pressing registration.

---

<sup>1</sup> <https://www.figma.com>

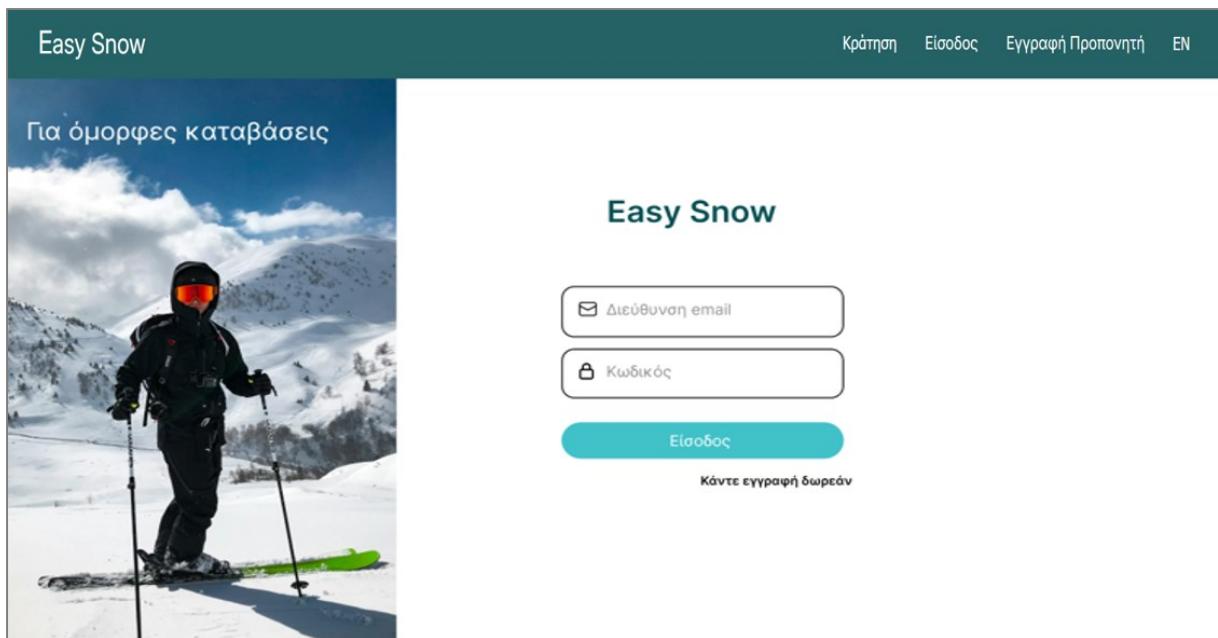


Image 25. User login to the app.

### 5.3.3 Signup

When the user chooses the registration option, they are taken to the screen shown in figure 26, where they can choose to register as a coach or as a student. The user is then taken to the registration page shown in Figure 27 to fill in their details. Specifically, he/she fills in his/her full name, e-mail address, telephone number and password. However, this page covers the first functional requirement of Table 3, which is user registration.

In fact, it is worth mentioning that many users of the app may be tourists and do not know Greek. For this reason, in the upper right part of the navigation menu there is an option to change the language of the application to English, which is the fourteen functional requirement of Table 3.

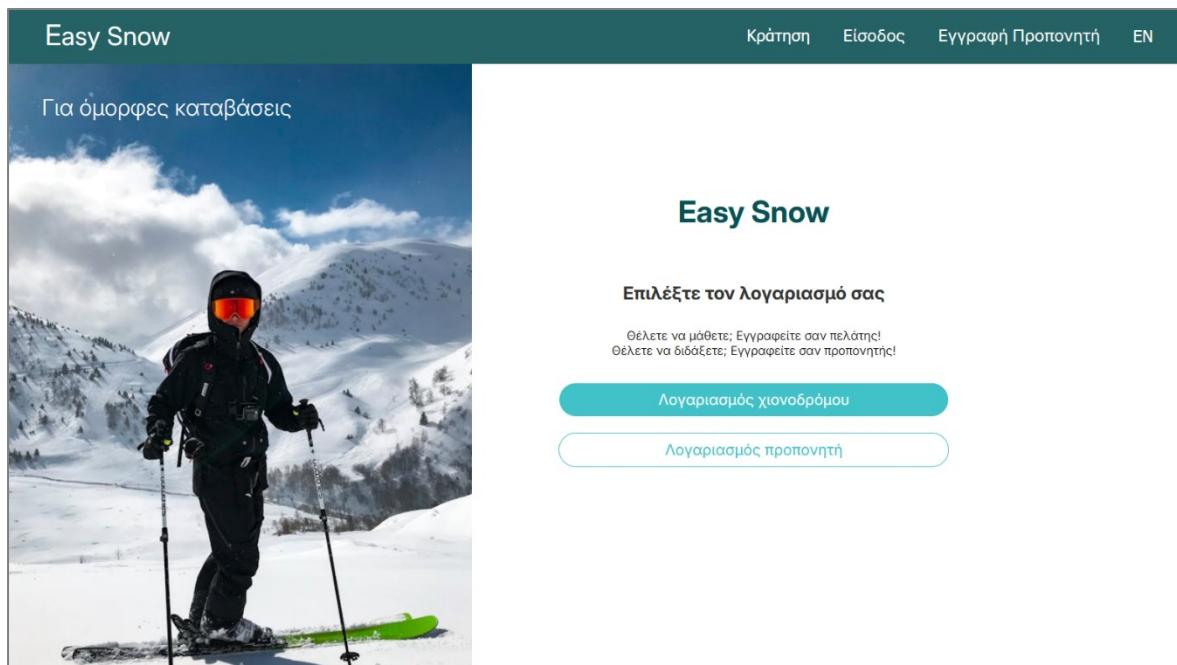


Image 26. Select a membership.

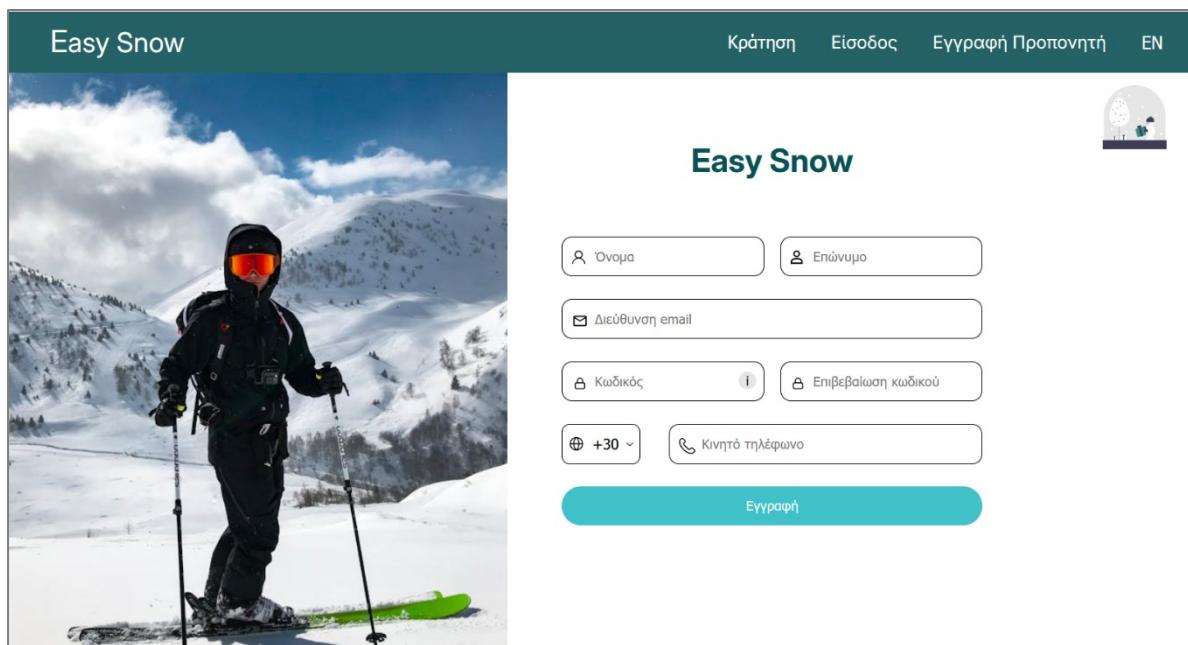


Image 27. Register a new user.

### 5.3.4 Selecting Course Parameters

In fact, the user can press book on the home screen or from the navigation menu and the lesson configuration window shown in figure 28 opens. The user then fills in the basics for the courses he wants to plan, such as the ski resort, the dates, the sport and the number of participants. In fact, these steps are the same as those that emerged during the hierarchical analysis for the planning of a course, which are shown in Figure 21.

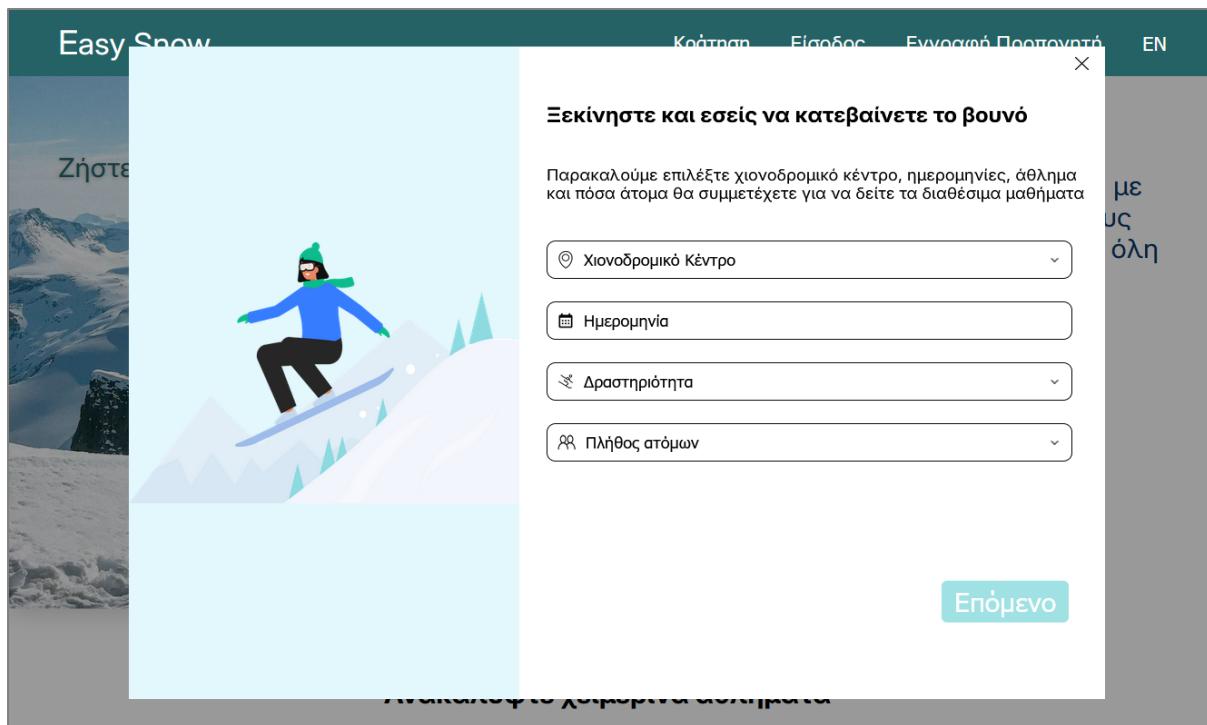


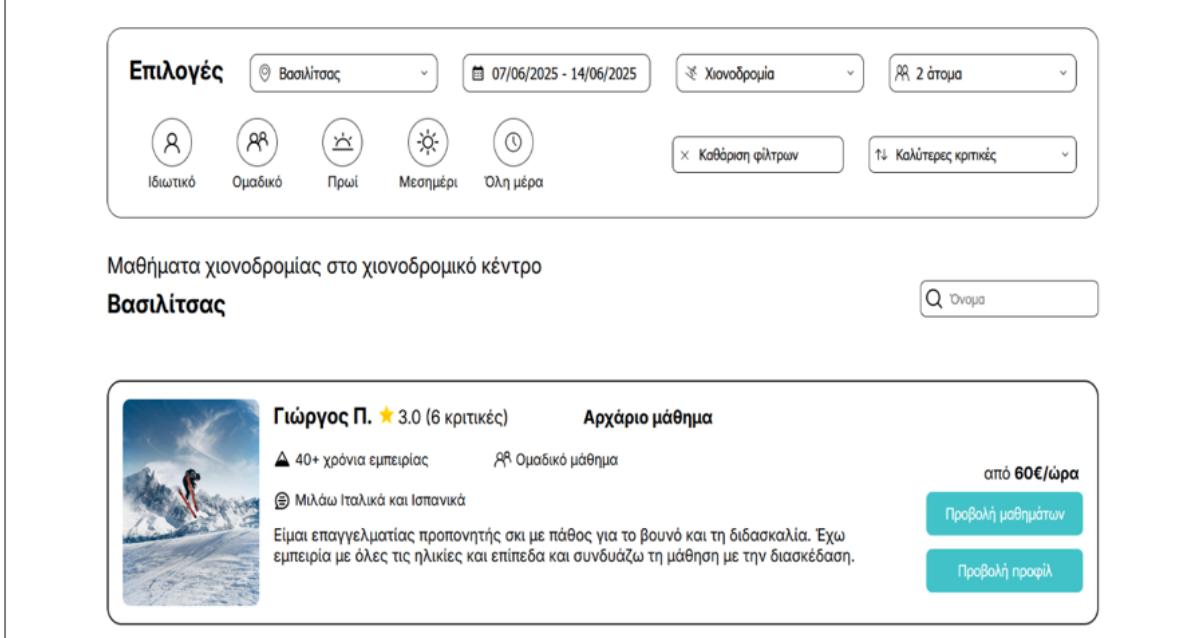
Image 28. Lesson planning.

### 5.3.5 Selection of courses

The course selection menu is shown in figure 29. First of all, we notice that this is a multi-criteria menu, i.e. the user can select several filters at the same time and the results are immediately displayed on the page. Specifically, the user can choose the lesson to be private, i.e. that only one group can participate in it, or group, i.e. that many unknown people can participate. Also, the user can choose various preferences, such as the time of the lesson being morning or noon, or the lesson lasting all day. In fact, the user has the option to sort the results by various criteria, such as coaches' evaluations or cost. In addition to these, the user can search for the name of a specific coach who may, for example, have scheduled a lesson with him or her in the past and was happy. However, page of Figure 29 covers the third functional requirement of Table 3.

Also, at the top of the page there is a progress bar that informs the user which steps they have already taken and which ones remain. Specifically, we notice that in this step the user has already selected the desired dates and is now in the course selection step. It will then be taken to the overview of the selected courses and finally to the payment page.

About coaches, the user sees their ratings and qualifications, such as the experience they have and the languages they know. In addition, users are informed about the cost of lessons per hour and whether the lessons are group or private. In fact, users can see details about the courses and times by tapping "View Courses" and, in addition, by tapping "View Profile" they are taken to the coaches' profile page.



The screenshot shows the course selection interface. At the top, there are four dropdown menus: 'Βασιλίτσας' (Vasilitsa), '07/06/2025 - 14/06/2025', 'Χιονοδρομία' (Snowboard Run), and '2 άτομα' (2 people). Below these are five filter icons: 'Ιδιωτικό' (Private), 'Ομαδικό' (Group), 'Πρωί' (Morning), 'Μεσημέρι' (Afternoon), and 'Όλη μέρα' (All day). To the right are two buttons: 'Καθάριση φίλτρων' (Clear filters) and 'Καλύτερες κριτικές' (Best reviews). A search bar with placeholder 'Όνομα' (Name) is also present. The main content area displays a course card for 'Γιώργος Π.' (George P.) with a yellow star rating of 3.0 (6 reviews). The card includes a photo of a skier, the title 'Αρχάριο μάθημα' (Beginner lesson), and details: '▲ 40+ χρόνια εμπειρίας' (▲ 40+ years of experience), 'Ομαδικό μάθημα' (Group lesson), and 'Μιλάω Ιταλικά και Ισπανικά' (I speak Italian and Spanish). It also states: 'Είμαι επαγγελματίας προπονητής σκι με πάθος για το βουνό και τη διδασκαλία. Έχω εμπειρία με όλες τις ηλικίες και επίπεδα και συνδυάζω τη μάθηση με την διασκέδαση.' (I am a professional ski instructor with passion for the mountain and teaching. I have experience with all age groups and levels and combine learning with fun.). The price 'από 60€/ώρα' (from 60€/hour) is shown in a green button, along with 'Προβολή μαθημάτων' (View lessons) and 'Προβολή προφίλ' (View profile).

Image 29. Course selection, top of page.

Of course, the course selection page is large and not all of it is captured in one image. For this reason, the page appears in two images with Figure 29 being the upper part and Figure 30 being the bottom. In fact, we see that at the bottom of the page there are user reviews about the coaches for the ski resort that the user has chosen. Specifically, each review contains how many "stars" users rated the coach, the comments they had and the hours they took a lesson with him.

The screenshot shows a course selection interface for 'Easy Snow'. At the top, there's a navigation bar with 'Easy Snow' on the left, and 'Κράτηση', 'Μαθήματα', a user icon, and 'EN' on the right. Below the navigation is a search bar containing the text 'εμπειρία με όλες τις ηλικίες και επίπεδα και συνδυάζω τη μάθηση με την διασκέδαση.' To the right of the search bar is a blue button labeled 'Προβολή προφίλ'. Below the search bar, there are navigation arrows and a page number '1'.

**Κριτικές Χρηστών**

**Γιώργος Γ.** ★★★★★  
02/06/2025 - Χιονοδρομία, Βασιλίτσα  
Κατανοητές οδηγίες και ωραία διάρρομη  
 Ο/Η Γιώργος Γ. έκανε κράτηση 6 ώρες με τον/την Γιώργος Π.

**Γιώργος Γ.** ★★★★★  
04/06/2025 - Χιονοδρομία, Ανήλιο  
Πέρασα πολύ καλά!!  
 Ο/Η Γιώργος Γ. έκανε κράτηση 7 ώρες με τον/την Γιώργος Π.

**Γιώργος Γ.** ★★★★☆  
04/06/2025 - Χιονοδρομία, Ανήλιο  
Ήταν πολύ ωραία  
 Ο/Η Γιώργος Γ. έκανε κράτηση 7 ώρες με τον/την Γιώργος Π.

**Γιώργος Γ.** ★★★☆☆  
02/06/2025 - Χιονοδρομία, Βασιλίτσα  
Πολύ ωραίο  
 Ο/Η Γιώργος Γ. έκανε κράτηση 6 ώρες με τον/την Γιώργος Π.

Image 30. Course selection, bottom of page.

### 5.3.6 View instructor profiles

For each lesson in Figure 29 the user sees various elements of the coach. Of course, if the user wants to know more about the coach, he can see his profile, as shown in figure 31. Among other things, on this page the user can be informed about all the sports taught by the coach, his cancellation policy and other information related to the coach. In addition, the user can read reviews written by previous students of the coach. Also, the user can contact the coach through an email form. This is useful for users who may want to ask the coach something that is not mentioned in their profile. In addition, page of Figure 31 covers the functional requirements ten and eleven of Table 3.

**Νίκος Μ.**

★ 1 ( 2 κριτικές )  
▲ 30 χρόνια εμπειρίας

Επικοινωνία

⌚ Κάνω μαθήματα σε Παρνασσό ⌐ Μιλάω Ελληνικά, Αγγλικά και Γαλλικά

🕒 Διδάσκω χιονοδρομία

Η πολιτική ακύρωσης είναι δωρεάν  
μέχρι 7 ημέρες πριν το μάθημα

▲ 30 χρόνια εμπειρίας

Είμαι πιστοποιημένος εκπαιδευτής σκι με πάνω από 30 χρόνια εμπειρίας σε ελληνικά και διεθνή χιονοδρομικά κέντρα. Εξειδικεύμαται στην εκπαίδευση παιδιών και ενηλίκων όλων των επιπέδων, από αρχάριους μέχρι προχωρημένους. Η διδασκαλία του σκι αποτελεί για μένα πάθος και στόχος μου είναι κάθε μαθητής να νιώσει αυτοπεποίθηση και ασφάλεια στο βουνό, ενώ διασκεδάζει και βελτιώνει την τεχνική του. Μιλάω ήππαιστα αγγλικά και γαλλικά και προσφέρω ιδιαίτερα και ομαδικά μαθήματα με προσφρούμασμένη προσέγγιση ανάλογα με τις ανάγκες του κάθε μαθητή.

### Κριτικές Χρηστών

★★★★★

**ΚΩΝΣΤΑΝΤΙΝΟΣ Σ.**

06/06/2025 - Χιονοδρομία,  
Παρνασσός

Πολύ καλός προπονητής

O/H KONSTANTINOS S

★★★★★

**ΚΩΝΣΤΑΝΤΙΝΟΣ Σ.**

07/06/2025 - Χιονοδρομία,  
Παρνασσός

Πολύ καλός προπονητής

O/H KONSTANTINOS S

★★★★★

**ΚΩΝΣΤΑΝΤΙΝΟΣ Σ.**

08/06/2025 - Χιονοδρομία,  
Παρνασσός

Πολύ καλός προπονητής

O/H KONSTANTINOS S

★★★★★

**ΚΩΝΣΤΑΝΤΙΝΟΣ Σ.**

09/06/2025 - Χιονοδρομία,  
Παρνασσός

Βελτιώθηκε πολύ η τεχνική μου!

O/H KONSTANTINOS S

Image 31. Coach profile.

### 5.3.7 View Course Hours

For the courses shown in figure 29, the user can see the available hours by clicking on "View Courses". In fact, as shown in figure 32, the user sees all the available courses of the coach for the dates he is interested in and can choose more than one course according to his preferences. In addition, the user must choose their level in the sport, i.e., whether they are beginner, intermediate, or advanced. Also, the user can see the meeting point set by the coach. In fact, because many students are not familiar with the slopes and the various locations on the mountain, the coach may attach a photo of a map of the ski resort with the meeting point marked to make it easier for the skiers to understand where they are. Of course, page of figure 32 covers the fourth functional requirement regarding the scheduling of courses with coaches.

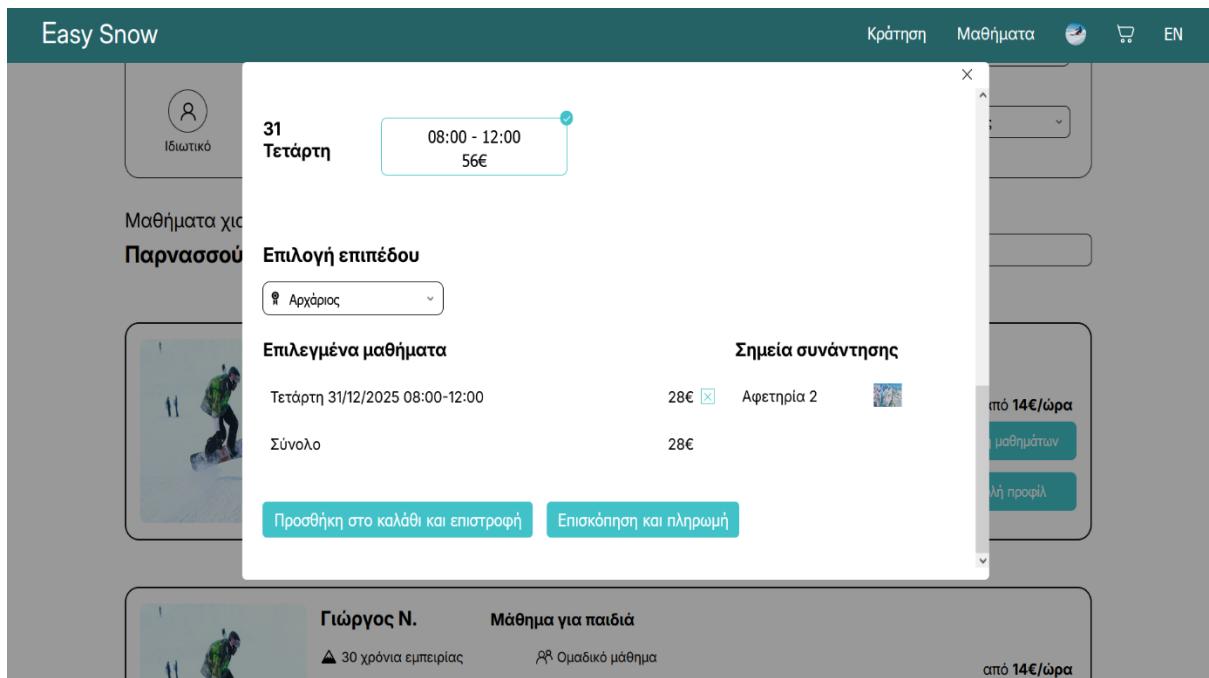


Image 32. View available courses.

### 5.3.8 Overview of selected courses

The user, when he has selected the courses he wants, can see their details in the overview menu shown in figure 33. Specifically, it sees data about the coach, the course, the dates, the participants and the meeting point. However, if the user has selected a course that they do not want, they can easily remove it. Of course, in case the user has forgotten to add a course he wishes, he can return to the course selection menu and select it. Finally, when the user is satisfied with the selected courses, they can go to the payment menu to complete the transaction.

### 5.3.9 Payment

The user to make a payment is taken to the payment menu shown in Figure 34. In fact, the user fills in his card details, such as the name of the cardholder, the card number, the expiration date and the card code. In fact, if the details are correct, he chooses payment and the transaction is completed.

### 5.3.10 Overview of Student Scheduled Courses

The student, after logging in to his account, can see his scheduled lessons and his course history as shown in figure 35. In fact, the scheduled courses show the courses that have not yet been completed, while the course history shows the courses that have been completed. For each lesson, the user sees all relevant data, such as the date, time, and coach details. In fact, the user sees the coach's experience, the languages they speak, the cancellation policy and the coach's mobile phone, which is only available to the user if they have scheduled a lesson with the coach. Also, the user sees the details of the lesson, such as the type of sport and whether the lesson is private or

group. Also, the user sees the meeting point along with a photo of a map where the meeting point is marked. In addition, the user has the ability to contact the coach through a contact form to ask him any questions he may have. In fact, the user can also cancel a lesson depending on the cancellation policy chosen by the coach. Also, the user can write a review for the courses he has already taken. In conclusion, page of Figure 35 covers the functional requirements seven, nine, twelve, thirteen, and fifteen of Table 3.

However, on the left side of image 35 you can see the student's navigation menu when he logs in to his account. In particular, the student can still go to their profile page. Also, with the collapse option at the bottom of the menu, the menu is made smaller and only the icons are shown so that it takes up less space.

The screenshot shows a mobile application interface for 'Easy Snow'. At the top, there is a navigation bar with the title 'Easy Snow' on the left and links for 'Κράτηση', 'Μαθήματα', a user icon, a shopping cart icon, and 'EN' on the right. Below the navigation bar is a horizontal menu bar with four items: 'Επιλογή ημερομηνιών' (with a calendar icon), 'Επιλογή μαθημάτων' (with a cursor icon), 'Επισκόπηση' (with a magnifying glass icon), and 'Πληρωμή' (with a credit card icon). A teal-colored button labeled '← Επιστροφή' is located below the menu bar. The main content area is titled 'Επιλεγμένα μαθήματα' (Selected Courses). It displays a course card for 'Νίκος Μ.' with a rating of ★ 5.0 (2 κριτικές). The course name is 'Traversa'. The course details include: 'Παρνασσός', 'χιονοδρομία', 'Ομαδικό μάθημα', and '30 χρόνια εμπειρίας'. To the right of these details are 'Πληροφορίες συμμετεχόντων': '2 άτομα' and 'Αρχάριος'. Below the course details, there are two sections: 'Επιλεγμένα μαθήματα' (Selected Courses) showing a session from 'Τρίτη 16/12/2025 13:00-15:00' for 24€ with 2 spots available, and 'Σημεία συνάντησης' (Meeting Points) showing a location with a pin icon. A teal button labeled 'Χ Αγοράστηκε μαθήματος' is located next to the meeting point. At the bottom right of the main content area, the total amount 'Σύνολο 24€' is displayed.

Image 33. Overview of selected courses.

Επιλογή ημερομηνιών      Επιλογή μαθημάτων      Επισκόπηση      Πληρωμή

← Επιστροφή

**Στοιχεία Κάρτας**

Όνομα Κατόχου	Αριθμός Κάρτας
Ημερομηνία Λήξης	CVV

**Σύνολο 140€**

Πληρωμή

Image 34. Payment menu.

Επιλογή ημερομηνιών      Επιλογή μαθημάτων      Επισκόπηση Μαθήματα EN

Προφίλ  
Μαθήματα  
Αποσύνδεση

### Προγραμματισμένα Μαθήματα

**Γιάννης Μ.** ★ 4.9 (27 κριτικές)

  
Παρνασσός  
Σκι  
7 χρόνια εμπειρίας  
Μιλάω Ελληνικά και Αγγλικά

Ημερομηνίες  
Δευτέρα 05/01/2025  
Όλη μέρα (8:00- 15:00)  
Τρίτη 06/01/2025  
11:00- 13:00  
Σύνολο

100€  
40€  
140€

Ιδιωτικό μάθημα  
 Η πολιτική ακύρωσης είναι μέχρι 7 ημέρες πριν το μάθημα

Σημείο Συνάντησης  
Άνω σαλέ 

**Πληροφορίες Συμμετεχόντων**  
♂ 4 Ατόμα ♀ Αρχάριο επίπεδο  
Τηλέφωνο Προπονητή  
6999999999

**Επικοινωνία**  
 Ακύρωση μαθήματος

### Ιστορικό Μαθημάτων

**Γιώργος Δ.** ★ 4.7 (34 κριτικές)

  
Παρνασσός  
Σκι  
7 χρόνια εμπειρίας  
Μιλάω Ελληνικά και Αγγλικά

Ημερομηνίες  
Δευτέρα 05/01/2025  
Όλη μέρα (8:00- 15:00)  
Τρίτη 06/01/2025  
11:00- 13:00  
Σύνολο

100€  
40€  
140€

Ιδιωτικό μάθημα  
 Η πολιτική ακύρωσης είναι μέχρι 7 ημέρες πριν το μάθημα

Σημείο Συνάντησης  
Άνω σαλέ 

**Πληροφορίες Συμμετεχόντων**  
♂ 4 Ατόμα ♀ Αρχάριο επίπεδο  
Τηλέφωνο Προπονητή  
6999999999

**Κριτική**

Σύμπτυξη

Image 35. Overview of ski lessons.

### 5.3.11 Student profile

The skier can see his details on his profile, as shown in figure 36. In fact, he can see his name, email address and mobile phone. He can also modify the corresponding fields in case he has declared something incorrectly or has changed something. He can also click "Change Password" to change his password. Therefore, the profile page covers the functional requirements five and six of Table 3.

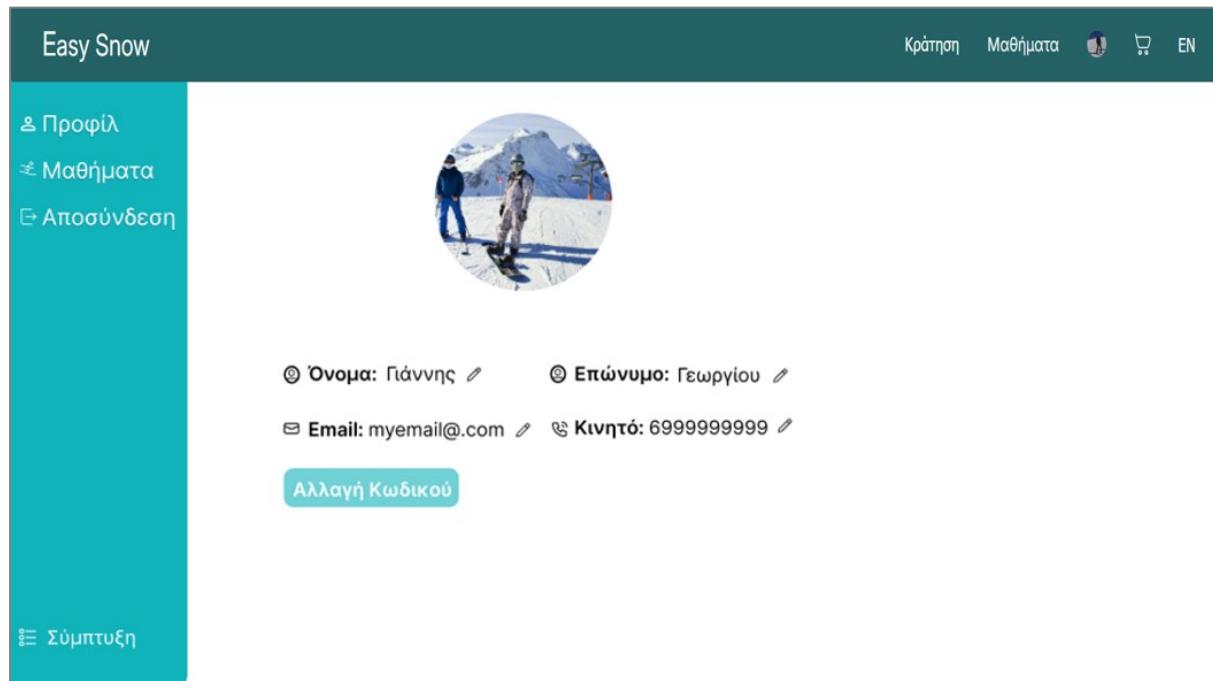


Image 36. Skier profile.

### 5.3.12 Managing Coach Courses

The navigation menu when a user who is a coach logs in is different from that for a skier who is a student and is shown on the left side of image 37. Specifically, in addition to his profile, the coach can see his schedule, manage his lessons and see some statistics. In fact, in the course management menu, the coach can initially specify the strengths of the meeting points for his courses, as shown at the top of image 37. It can then create a course. This is done by filling in the ski resort and the dates that the lesson wants to take place. In fact, the coach can exclude some days of the week, e.g. Monday, because he may have an obligation on that day. Therefore, a coach can, for example, choose to do a lesson throughout December, except on Mondays, because he may have another obligation. Also, the coach determines the time of the lesson, whether the lesson lasts all day, whether it is private or group, the type of sport, the maximum number of

participants and chooses one of the meeting points he has already defined. Finally, the coach determines the price per hour and proceeds to the creation of the course.

The screenshot shows the top portion of a web application for managing ski courses. The header "Easy Snow" is at the top left, with a dark teal sidebar on the left containing navigation links: Προφίλ, Διδασκαλίες, Πρόγραμμα, Στατιστικά, Αποσύνδεση, and Σύμπτυξη. The main content area has a light gray background. It features a section titled "Σημεία Συνάντησης" (Meeting Points) showing two locations: "Καλάβρυτα Άνω σαλέ" and "Καλάβρυτα Κάτω σαλέ", each with a small map icon. Below this is a button "+ Προσθήκη σημείου συνάντησης". Underneath is a section titled "Δημιουργία Μαθήματος" (Create Lesson). This section contains several input fields and dropdown menus: "Χιονοδρομικό Κέντρο" (Parassos), "Ώρα έναρξης- Ήμερη" (8:00 AM - 12:00 PM), "Μέγιστο πλήθος συμμετεχόντων" (4 people), "Ημερομηνία Έναρξης" (05/01/2025), "Το μάθημα διαρκεί όλη μέρα" (Yes/No), "Επιλογή δυνατών σημείων συνάντησης" (Epilogi), "Ημερομηνία Λήξης" (05/01/2025), "Τύπος Μαθήματος" (Individual/Group), "Πολιτική Ακύρωσης" (Politiki Akyrwsis), "Να εξαιρεθεί κάποια μέρα" (Exclude certain days), "Αθλημα" (Athletika), "ΣΚΙ" (Ski), "Τιμή ανά ώρα" (Price per hour: €40), and a final button "+ Δημιουργία" (Create).

Image 37. Coach Course Management, Top Page.

In fact, the instructor's course management page is long and has been divided into two images where the bottom of the page is shown in image 38. In particular, the coach can see the lessons he has created in a concentrated way, so that he knows which days and times he has a lesson. However, if for reasons of force majeure the coach needs to cancel a lesson, he can do so by pressing the corresponding button. Of course, if people are already enrolled in the course, they must be informed by the coach about the cancellation. However, one reason that a coach may need to cancel all lessons that are done on the dates he has stated on this page, is in case he made a mistake when creating a lesson and sees it immediately afterwards, in which case he should be able to immediately cancel all lessons that were created. In fact, page of Figure 37 covers functional requirement two of Table 3.

Easy Snow
Πρόγραμμα Διδασκαλίες Στατιστικά
 EN

- [Προφίλ](#)
- [Διδασκαλίες](#)
- [Πρόγραμμα](#)
- [Στατιστικά](#)
- [Αποσύνδεση](#)
- [Σύμπτυξη](#)

Χιονοδρομικό Κέντρο

Ώρα έναρξης- Ήρα Λήξης

Μέγιστο πλήθος συμμετεχόντων

Ημερομηνία Έναρξης

Το μάθημα διαρκεί όλη μέρα

 Ναι  Οχι

Επιλογή δυνατών σημείων συνάντησης

Ημερομηνία Λήξης

Τύπος Μαθήματος

 Ιδιωτικό  Ομαδικό

Πολιτική Ακύρωσης

(i)

Να εξαιρεθεί κάποια μέρα

Αθλημα

Δυνατότητα ακύρωσης μέχρι  
7 ημέρες πριν το μάθημα

Τιμή ανά ώρα

+ Δημιουργία

### Δημιουργημένα Μαθήματα

Χιονοδρομικό Κέντρο

Ώρα έναρξης- Ήρα Λήξης

Μέγιστο πλήθος συμμετεχόντων

Ημερομηνία Έναρξης

Το μάθημα διαρκεί όλη μέρα

 Ναι  Οχι

Επιλογή δυνατών σημείων συνάντησης

Ημερομηνία Λήξης

Τύπος Μαθήματος

 Ιδιωτικό  Ομαδικό

Πολιτική Ακύρωσης

(i)

Να εξαιρεθεί κάποια μέρα

Αθλημα

Δυνατότητα ακύρωσης μέχρι  
7 ημέρες πριν το μάθημα

Τιμή ανά ώρα

x Ακύρωση

Image 38. Manage Coach Courses, Bottom Section of Page.

### 5.3.13 Instructor program

The coach can see in detail the program he has for one day on the program overview page shown in figure 39. Initially, he selects the day he is interested in and then the scheduled lessons are displayed below. In fact, the lessons indicate, among other things, the time, the meeting point, the number of participants and the contact details of the students. So that if there is difficulty in meeting them, it is easy to communicate. Therefore, page of Figure 39 covers requirement eight of Table 3 of the operational requirements.

The screenshot shows a mobile application interface for 'Easy Snow'. The top navigation bar includes links for 'Πρόγραμμα' (Program), 'Διδασκαλίες' (Lessons), 'Στατιστικά' (Statistics), and language options 'EN' and a user icon. On the left, a vertical sidebar lists menu items: 'Προφίλ' (Profile), 'Διδασκαλίες' (Lessons), 'Πρόγραμμα' (Program) which is highlighted in blue, 'Στατιστικά' (Statistics), and 'Αποσύνδεση' (Logout). The main content area displays a 'Πρόγραμμα' (Program) titled '24/12/2025'. It includes sections for 'Μαθήματα' (Lessons) and 'Στοιχεία μαθήματος' (Lesson details). Lesson details show a location in 'Αθλητικό Χιονοδρομία' at 'Αφετηρία 2' with a small map icon. The lesson date is 'Ημερομηνία Τετάρτη 24/12/2025' and the time is 'Ώρα 08:30-15:00'. The lesson type is 'Τύπος μαθήματος Ιδιωτικό'. Below this is a section for 'Στοιχεία μαθητών' (Student details) featuring a photo of a skier, the student's name 'ΚΩΝΣΤΑΝΤΙΝΟΣ ΣΤΑΥΡΟΠΟΥΛΟΣ', phone number '306951231212', email 'a@gmail.com', and experience level 'Αρχάριος'. A button at the bottom says 'Προσθήκη σημείωσης' (Add note).

Image 39. Coach program.

### 5.3.14 Instructor profile

The instructor can review and modify his details in his profile as shown in figure 40. In fact, the coach's details concern all the basic details of a user, such as first name, last name, email address and phone, but also many more. In particular, the coach fills in information about the ski resorts in which he is located, the sports he teaches, his experience, the foreign languages he knows, the cancellation policy of his lessons, his CV and some brief information that he wishes to appear in his lessons. In conclusion, the coach's profile page covers the sixteen functional requirement of Table 3.

Easy Snow

Πρόγραμμα Διδασκαλίες Στατιστικά EN

**Επιπρόσθετες πληροφορίες**

Επιλέξτε τα χιονοδρομικά κέντρα στα οποία διδάσκετε:

Ανηλίου	<input type="checkbox"/>	Βασιλίτσας	<input type="checkbox"/>
Βελουχίου	<input type="checkbox"/>	Ελατοχωρίου	<input type="checkbox"/>
Καιμακτσαλάν	<input type="checkbox"/>	Καλαβρύτων	<input type="checkbox"/>
Μαινάλου	<input type="checkbox"/>	Παρνασσού	<input checked="" type="checkbox"/>
Πηλίου	<input type="checkbox"/>	Πισοδερίου	<input type="checkbox"/>
Φαλακρού	<input type="checkbox"/>	3-5 Πηγάδια	<input type="checkbox"/>

Επιλέξτε τις γλώσσες που μιλάτε:

Ελληνικά	<input checked="" type="checkbox"/>
Αγγλικά	<input checked="" type="checkbox"/>
Γαλλικά	<input checked="" type="checkbox"/>
Ισπανικά	<input type="checkbox"/>
Ιταλικά	<input type="checkbox"/>
Γερμανικά	<input type="checkbox"/>

**Επιλέξτε τα αθλήματα που διδάσκετε:**

Χιονοδρομία	<input checked="" type="checkbox"/>
Χιονοσανίδα	<input type="checkbox"/>
Καθιστό σκι	<input type="checkbox"/>

**Επιλέτε την πολιτική ακύρωσής σας:**

Δεν επιτρέπεται η ακύρωση	<input type="checkbox"/>
Η ακύρωση είναι δωρεάν μέχρι 7 ημέρες πριν το μάθημα	<input type="checkbox"/>

**Ακύρωση** **Αποθήκευση**

**Ακύρωση** **Αποθήκευση**

Image 40. Excerpt from the coach's profile.

### 5.3.15 Instructor statistics

Finally, the coach has access to statistics about the lessons he took in a month, as shown in figure 41. Specifically, the coach can see per month his total income, income from private and group lessons, total teaching hours and teaching hours of private and group lessons. In fact, this page covers the functional requirement seventeen of Table 3.

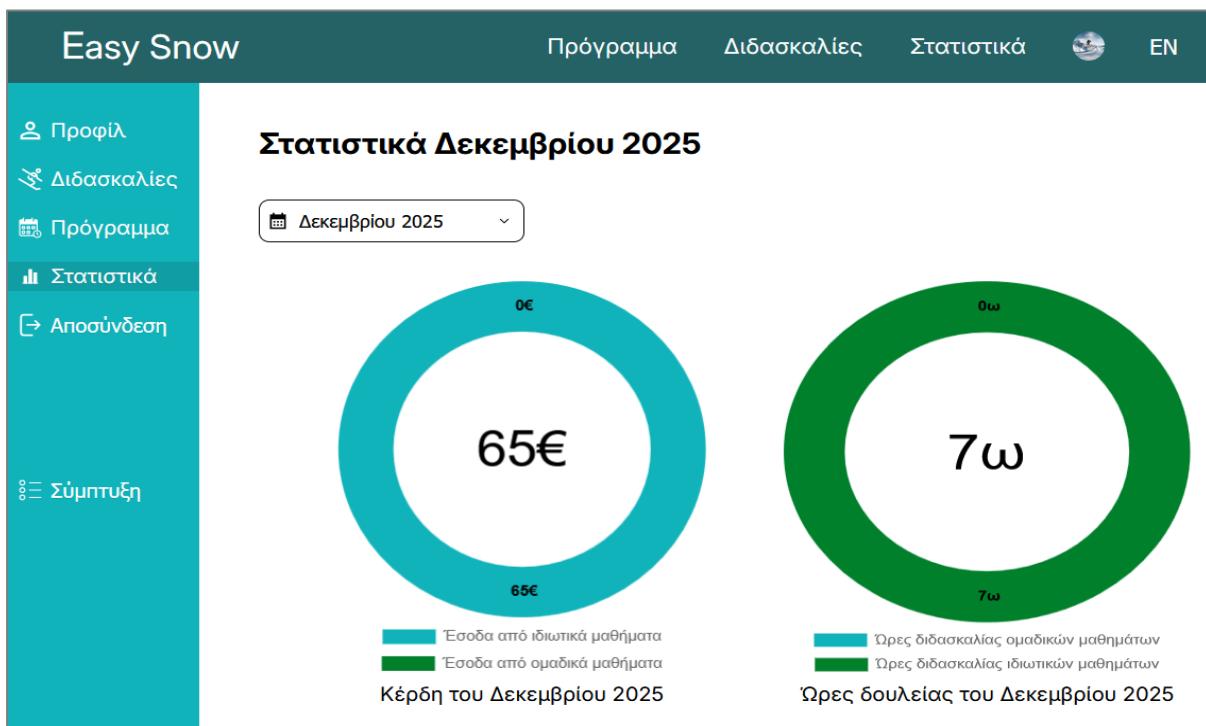


Image 41. Coaching statistics.

## 5.4 Evaluating prototypes with users

As we mentioned in section 5.1, one of the main benefits of the design process is that it allows us to evaluate prototypes early on and get feedback from standard users. Also, the evaluation according to the anthropocentric theory presented in section 2.3.3, must be done in all phases of the development of an application and not only at its end. However, the evaluation methods are divided into two categories (Avouris et al. 2018):

- **Formative Evaluation:** Formative evaluation aims to configure the final product and improve its functions. Therefore, it aims to identify specific errors and correct them.
- **Inconclusive evaluation:** The inconclusive evaluation decides on whether and how good the final product is, so it serves to confirm the correct design and implementation without focusing on specific points.

In this section, a formative evaluation is carried out, i.e. the aim is to configure and improve the functions of the application by correcting any errors that exist in the drawings. In fact, the method of the Talking Subject is used where we are in the same space as a user and ask him to externalize his thoughts to us as he performs a formal process with the application. Of course, the evaluation will also be done to a greater extent during the development of the application as users will be able to test the functions in greater detail than is done in the prototypes and at the same time they will be able to evaluate how well the application is designed (concluding evaluation).

### 5.4.1 Think aloud protocol

In this assessment method, users are asked to take representative actions with the application and externalize their thoughts (Lewis, 1982). In fact, before the evaluation begins, we tell users that their thoughts can be about anything related to the application and the process they carry out, e.g. it can be something they don't know how to do, something that confuses them or something they like.

For the purpose of the evaluation, a meeting was held in the same place with three typical users and they were asked to schedule a lesson with a coach of their choice. They were then asked to review their scheduled courses and check their profile details.

From the evaluation process, it emerged that users easily interacted with the app's components and understood what actions they needed to take to complete the above actions. However, some users pointed out that the user's profile seemed uncluttered, and others that the scribbles in the payment menu didn't match the rest of the design.

Therefore, based on these observations, we have redesigned the corresponding pages, which are shown below. In fact, the new user profile can be seen in Figure 42. Specifically, we added a blue color to the photo section and a simple sketch on the right.

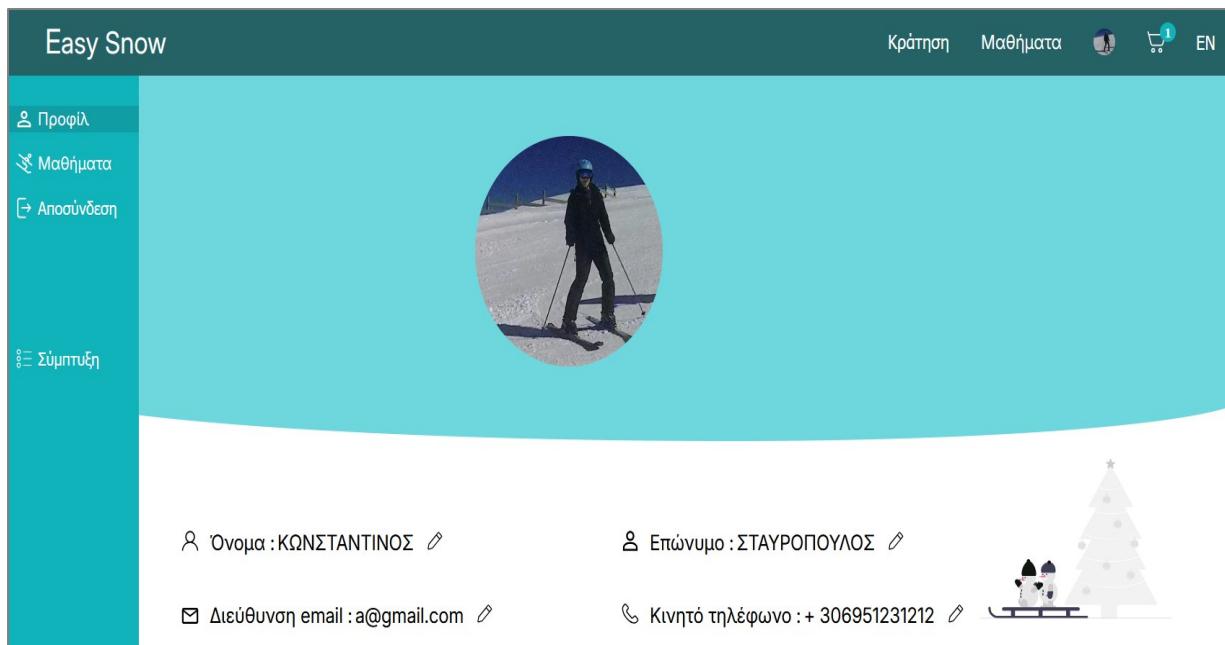


Image 42. Redesign user profiles.

The new payment page is shown in figure 43, where the payment form is placed on the right side of the screen and a scribble of credit cards on the left. Also, a glass with a snowman was placed over the overalls.

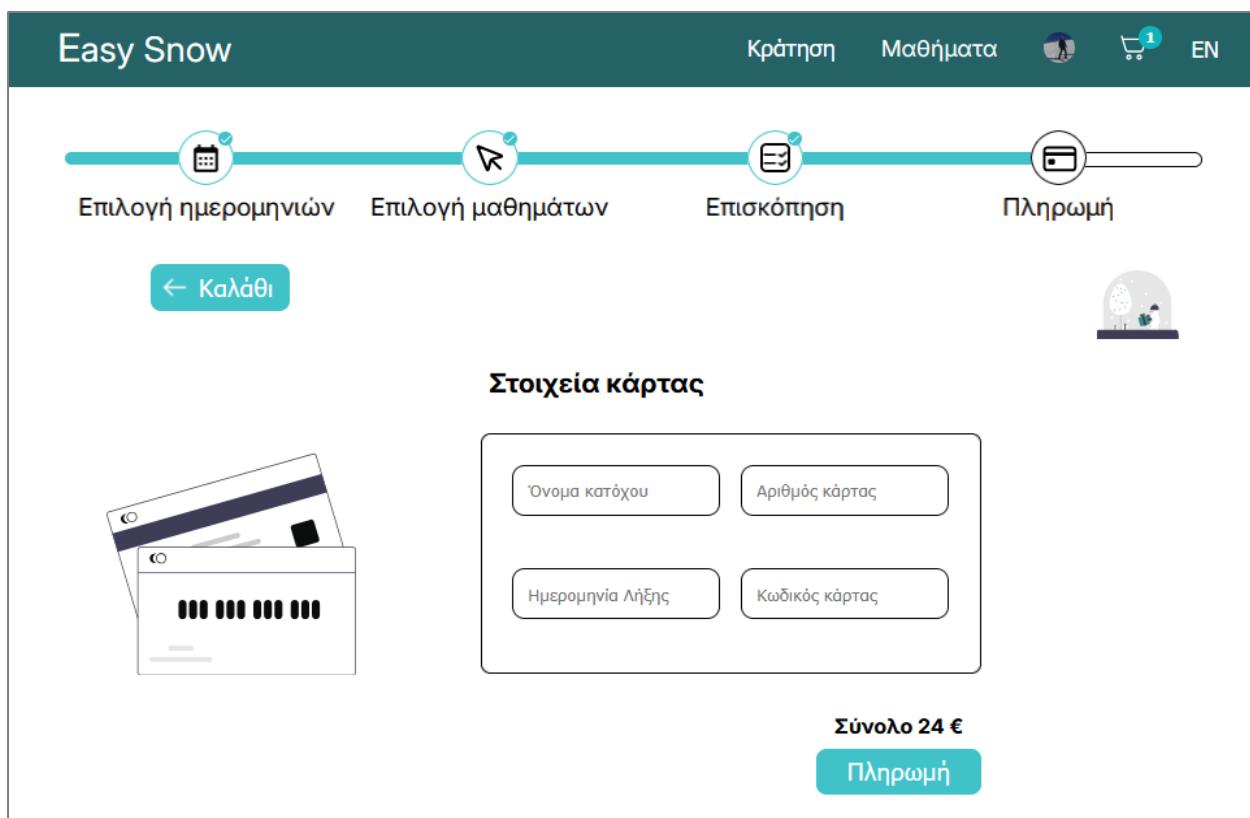


Image 43. Redesign a payment page.

## 5.5 Conclusions

In this chapter we first analyzed the importance of prototype design for the development of an application. Plans for the individual functions of the application were then created. The designs were then evaluated with the help of end-users. In particular, the Talking Subject Protocol was used, where users use the pages designed to carry out basic usage scenarios of the application, which were identified during the hierarchical analysis, and at the same time externalize their thoughts. The above process resulted in some aesthetic improvements to the user profile and payment page, which were incorporated into the final design. Therefore, the implementation of the application in code can now be done, which is carried out in the next chapter.

# 6 Implementation

This chapter analyses the technologies used to implement the application. Specifically, section 6.1 presents the general architecture of the application. Then, in section 6.2, the choices made for the development of the client (frontend) are justified, i.e. the part of the application that the end user sees. Furthermore, in section 6.3, the technologies selected for the server are mentioned. Section 6.4 also analyses the technologies used for the database. Finally, section 6.5 compares the estimated and actual implementation times of the application's functions.

## 6.1 Architecture

Figure 44 shows the general architecture of the Easy Snow app. Specifically, users interact with the Front-end, i.e. the code of the application that runs in the browser. The technologies used on the customer side are React and React Router, which are analyzed and explained below. Then, the server is deployed in a Node.js and Express.js environment that are also explained below. In fact, the server is essentially divided into three categories of functions, the routes, the controller functions, and the model functions. Routes correspond a url and a specific http method to a specific function of the controller that is responsible for processing the request. In fact, the controller function may need to communicate with the database. This communication requires some SQL commands, which are defined in model functions. Therefore, model functions access the PostgreSQL database and send the data back to the Controller, which sends the data to the user. Finally, the server communicates with external APIs and specifically EmailJS to send students' queries via email to the coaches.

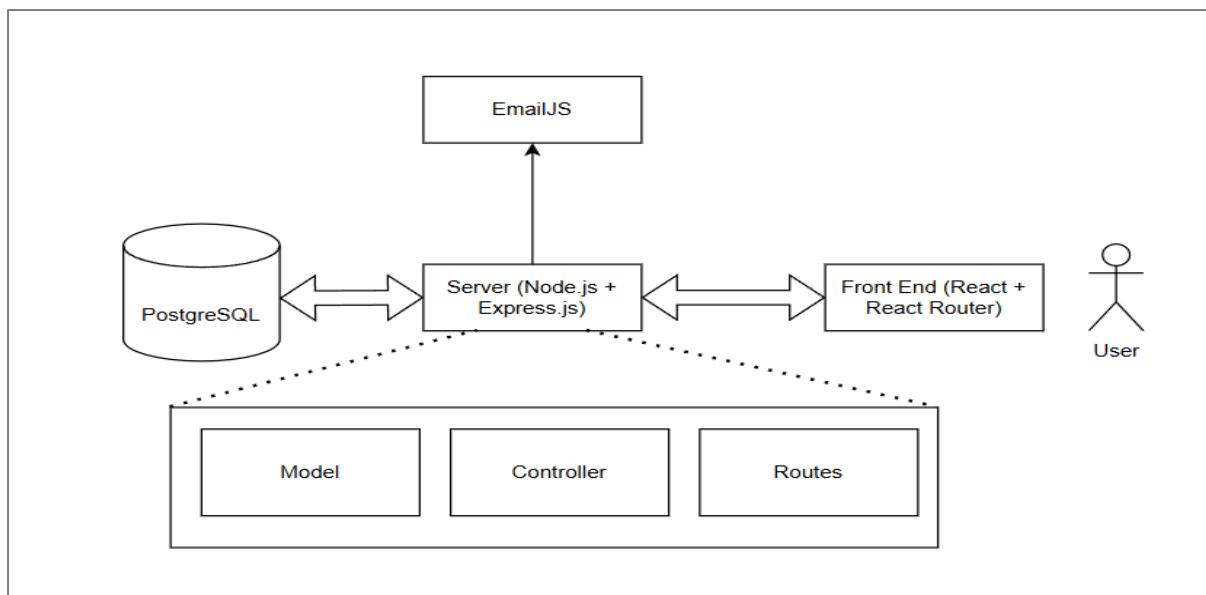


Image 44. Easy Snow application architecture.

## 6.2 Customer-Side Technologies

This section breaks down the technologies used on the client-browser side. It is noted that the customer is considered to be the end user of the application. In fact, three modern technologies for user interface development are initially compared, React, Angular and Vue. React, which was chosen to develop the code, is then further analyzed. It also describes the Vite that was used as React's bundler. Finally, reference is made to CSS commands and how they were used to create an application that adapts to different screen sizes (responsiveness).

### 6.2.1 Modern User Interface Development Technologies

Vue, React and Angular are three of the most popular and modern JavaScript frameworks and libraries used for the development of User Interfaces (UI) and Single Page Applications (SPA). Although they have a common goal, they differ in their philosophy, architecture, and use.

#### Vue.js

Vue.js is a progressive JavaScript framework created by Evan You in 2014. The word "progressive" means that it can be introduced gradually into an application, without requiring a complete reconstruction of existing code. Vue focuses on simplicity, flexibility, and ease of learning. It offers a clean and reactive data binding method and has a component-based system that allows the reuse of data (Vue.js, e.g.).

In fact, Vue is particularly popular with small and medium-sized development teams, as it provides excellent documentation, ease of learning, and lean syntax. Its community is quite large and is supported by a rich ecosystem of tools and extensions, such as Vue Router and Vuex.

#### React.js

React is a JavaScript library developed by Meta, the company that created Facebook, in 2013 (Meta, e.g.). It has as its main goal the creation of reusable UI components and the efficient management of the state of applications. Of course, React's key innovation is the Virtual DOM, which allows the real DOM to be efficiently updated when the data changes.

React uses JSX syntax, which allows HTML to be embedded within JavaScript code. The library focuses only on the "View" of the Model-View-Controller (MVC) architecture, so it is often used in conjunction with other libraries or frameworks for complete solutions. It is particularly popular in large and dynamic applications and is widely used by technology companies worldwide.

#### Angular

Angular is a complete TypeScript-based framework developed and supported by Google. It was first released in 2016 (as the successor to AngularJS) and offers a comprehensive application development solution, including routing, state management, forms, and dependency injection.

Angular is based on component architecture and makes extensive use of decorators, modules and services (Angular, e.g.). Although it has a more complex structure and harder to learn than Vue or React, it does provide powerful tools for building large, scalable applications.

## 6.2.2 React.js

For the Easy Snow application we have chosen to use React because we seek to make a Single Page Application, so Angular is considered more complex than we need and Vue relatively simpler. In the continuation of this section, React and its mechanism are analyzed in depth.

### React Operating Mechanism

React is one of the most widely used tools for creating modern interfaces, and its operating mechanism is based on the following properties:

- **Virtual Dom:** React uses a virtual DOM (Virtual DOM), which is a lightweight copy of the actual DOM. When the state of a component changes, React first updates the Virtual DOM and then compares the changes to the actual DOM, applying only the necessary updates (Deshpande, 2025).
- **JSX (Javascript XML):** React uses JSX, a syntax that allows HTML writing within JavaScript. This makes the code more readable and expressive, making it easier to develop user interfaces.
- **Props and State:** Props are properties that are transferred from a parent component to a child component, allowing components to be reused and configured. The state of an element represents its internal state and can change over time, causing the component to re-render when it changes. A typical example is a form element, whose status changes every time a button is pressed and the form becomes visible or invisible.

### Advantages of React

React requires a lot of familiarity and learning to understand its operating mechanism. Of course, the effort it takes is balanced by the various benefits it offers. Specifically:

- **Single Page Application (SPA):** Perhaps the biggest advantage of React is that it allows you to create applications where all the code is only displayed within one page. This is possible with the component architecture mentioned in the previous section. Therefore, when the user wants to change pages, one could say that this is done by hiding one page and displaying another in the same window. Because of this mechanism, SPAs offer significant advantages that make them ideal for creating modern and efficient web applications. Initially, they provide a smoother and more interactive user experience, as the content loads dynamically without the need for a full page refresh, offering speed and a compact UI where only the elements that need to be changed are changed, and the entire page doesn't have to load from scratch. In addition, the necessary resources (HTML, CSS, JavaScript) load only once, reducing load time and bandwidth consumption. Additionally, SPAs reduce the load on the servers, as most of the processing takes place on the client's side, allowing more users to be served with the same resources (Erolin 2023).
- **Client Side Routing:** A key element of the architecture of application SPAs is client-side routing. In the traditional model of web pages, server side routing is used. That is, the user initially makes a request to the server for a page. Based on the url, the server understands what the user is asking for and loads the corresponding data from a database. The server then creates an html file, fills it in with the data and sends it to the

client along with other files (css, images, etc.). Finally, the client-browser draws the final page. For each subsequent navigation to another page, the client repeats the process by sending a new request to the server, which in turn repeats the above steps to send the corresponding page.

In contrast, in applications that use client-side routing, the server initially sends a single HTML file to the client, regardless of the URL path. This "shell" is enriched and modified via JavaScript that runs in the browser. As you navigate further through the application, JavaScript undertakes to update the displayed content, based on the data that has already been loaded and the current URL path. Also, if new data is needed, then it is retrieved by the server, without the server having to send a new html file. This approach is typical of Single-Page Applications (SPAs), which load once and use JavaScript to dynamically render the content of the various "pages". The basic flow includes an initial request from the client to the server, which responds with the basic HTML document and necessary elements (such as JavaScript and CSS files). The customer interprets the JavaScript code, which determines what content will appear depending on the path of the URL. In fact, during subsequent path changes, the application updates the browser history and the content of the page with new data (Djirdeh, 2024).

### Possible difficulties in using React

From the above analysis, it follows that in order to use React we will have to address some fundamental issues. Specifically:

- **No History** : As mentioned in the previous section, React is used to create Single Page Applications, and every time it needs to load a page, the server does not send a new html file, but only the necessary data. This does mean that the user's browser history does not change automatically and remains stable. That is, the navigation between pages is not recorded in the history. This is of course disastrous from a point of view usability and user experience, so it definitely needs to be addressed. As we will see below, we use a router-specific library, the React Router, to deal with it.
- **Long initial loading time** : A second disadvantage of SPAs stems from the fact that discussed in the previous section about the server sends to the customer's first request an html file containing all the code for all pages of the application and then the client simply makes requests to load the data of each page. This results in a very long initial loading time of the application, Which of course makes the user experience difficult. This will be dealt with by slicing the code sent so that on each page the server sends only the JSX code that the page needs.
- **Code overload** : As we explained in react, Components are based on their internal state and are rendered again every time it changes. This results in even the entire page being rendered again every time a button is simply pressed if the code is not developed correctly. For example, this will happen if the entire page is a component and we don't slice the code into separate components. That's why it's very important define our components in an optimal way and break the code into individual parts (Kent, 2019; Abramov, 2021; Developer Way, 2023).

## React Router

As explained above, Single Page Applications do not automatically change entries in the browser history, so a separate library is needed to be responsible for this. For this purpose, we use the React Router library (React Router, e.g.). Yes. This library offers a variety of features that we've leveraged to improve the user experience. In particular, it allows us to manage the browser history, to address the problem mentioned above regarding the delay that occurs when the entire application code is sent from the beginning and, finally, it allows us to load only one part of the page and make the page appear normally to the user, while at the same time loading another part of the page. This is very important in case a page is too large and there is some data that takes a long time to load and is needed, for example, only at the bottom of the page. Then, it makes sense for the rest of the page to load normally before they load that data and show a predefined item in place until they load. All of this has been implemented with the React Router library and is analyzed to a greater extent afterwards.

### Manage history

The React Router library allows us to define elements as links and forms that extend the corresponding traditional HTML elements. When one of the traditional components is clicked, then a request is automatically sent to the server and the change is recorded in the history. The corresponding components of the React Router prevent the initial request from reaching the server, and instead based on the request made, they activate an appropriate function that undertakes to load only the required data into the browser, without sending http code from the server. At the same time, a new entry is placed in the user's history. (Annie, 2023).

### Lazy loading

As we mentioned in section 6.2.2 **Possible difficulties in using React** One problem that we need to solve when using React is that the entire application code is sent on the first visit, resulting in a delay in the application loading. However, there is the lazy loading function of the React Router that allows the asynchronous loading of the non-critical parts of a path, in order to optimize the performance of the application through the technique of code-splitting. This means that data and components that are not necessary for the initial page load are only loaded when required, thus reducing the size of the initial package and speeding up the implementation (Web Dev Simplified, 2022). However, it is noteworthy that even in the case of code splitting, the jsx code and the data needed load in parallel, i.e. the server does not need to send the code first for the React Router to understand which data to load, as it is done at the same time.

### Asynchronous Data Loading

As mentioned above, sometimes it is desirable to load some mandatory data in order for the page to render without having to load all the data, because for example some of the data may only be required at the bottom of the page. For the above case, the React Router has the Suspense mechanism, which allows the page to load before loading the unnecessary data and in its place to display a predefined element, for example a cursor that indicates that something is loading (React router, e.g.). We have used this mechanism to load reviews and courses, as shown in images 45 and 46 respectively.

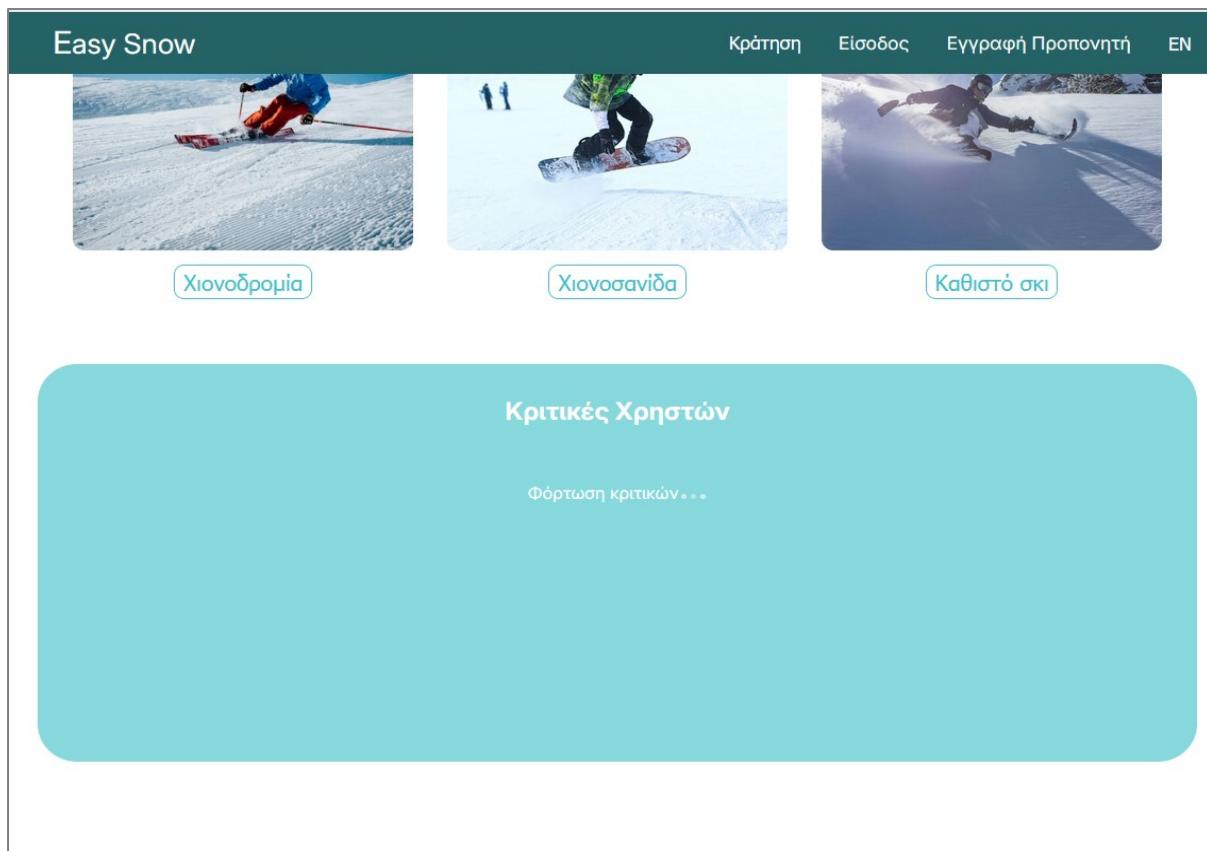


Image 45. Asynchronous loading of reviews.

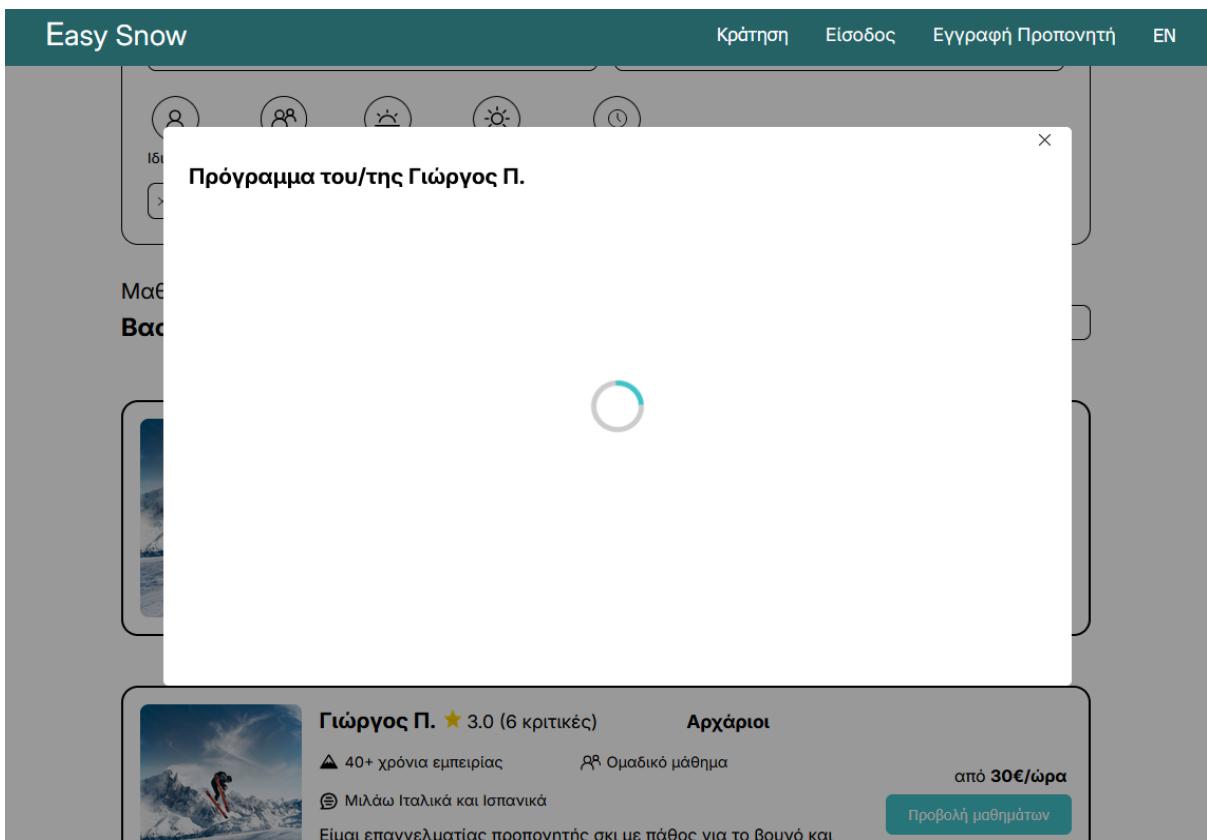


Image 46. Asynchronous loading of courses.

## Change UI during URL transitions

As we mentioned, React Router is a library responsible for routing on the browser side. Therefore, it can know when a URL or a user action is changing, e.g. the user can submit a form. That's why React Router gives us the ability to change the UI when the user takes an action. This change allows the user to know that indeed the action they took is taking place. The most typical example that this UI change has been leveraged in the app is when navigating through different course pages, as shown in Figure 47. Also in the menu of both the student and the coach, the selected page has a different color from the others (image 48). When the user switches pages, the app shows the new page to be selected before it loads. This, in addition to informing the user about his action, creates a feeling of immediacy at the same time.

The screenshot displays a mobile application interface for a snowboarding course platform. At the top, there is a navigation bar with the title "Easy Snow" and language options "Κράτηση", "Είσοδος", "Εγγραφή Προπονητή", and "EN". Below the navigation bar are three course cards, each featuring a profile picture of a snowboarder, the name of the instructor, their age and gender, their location, a brief description, and two teal-colored buttons for "View Profile" and "View Courses".

- Γιώργος Ν.**  
30 χρόνια εμπειρίας  
Ιδιωτικό μάθημα  
Μιλάω Αγγλικά, Ελληνικά και Γαλλικά  
Είμαι επαγγελματίας σκι με πάθος για το βουνό και τη διδασκαλία. Έχω εμπειρία με όλες τις ηλικίες και επίπεδα και συνδυάζω τη μάθηση με την διασκέδαση.  
από 20€/ώρα
- Γιώργος Π.** ★ 3.0 (6 κριτικές)  
40+ χρόνια εμπειρίας  
Ομαδικό μάθημα  
Μιλάω Ιταλικά και Ισπανικά  
Είμαι επαγγελματίας προπονητής σκι με πάθος για το βουνό και τη διδασκαλία. Έχω εμπειρία με όλες τις ηλικίες και επίπεδα και συνδυάζω τη μάθηση με την διασκέδαση.  
από 12€/ώρα

< 1 2 >

Image 47. Change UI when navigating to a different page.

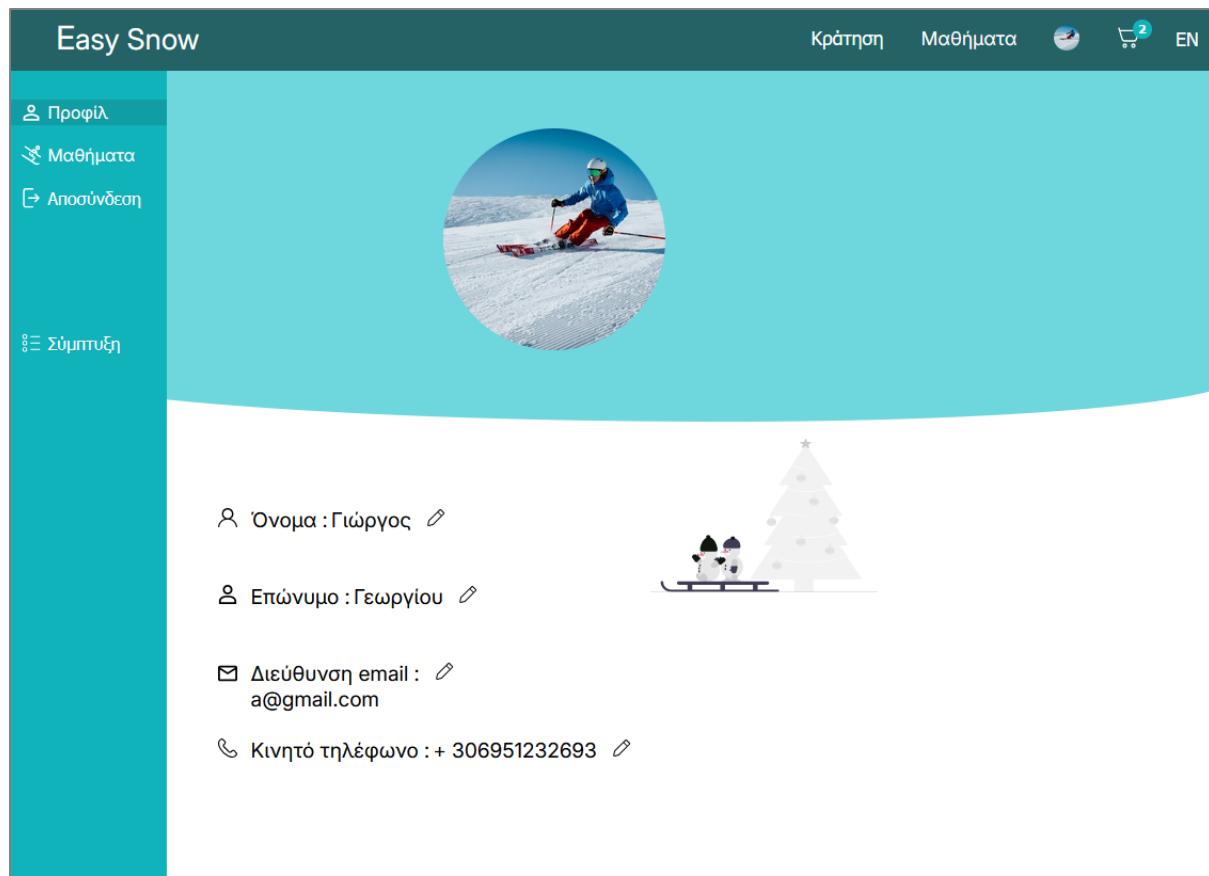


Image 48. The selected page has a different color in the menu.

### Optimistic UI

As shown in Figure 48, the user can change various details of his account, such as his first name, last name, etc. Typically, the user enters the new price he wants, e.g. his new phone and submits it. Until the new value is entered on the server, the user sees the old value in the corresponding field. However, a proper design technique that adds to the immediacy of the application is for the user to see the value they submit before it is entered on the server. If for some reason the registration on the server fails, then the old value on the user's page is automatically restored. This mechanism is known as the "optimistic" UI (Çite, 2025).

### Update the result of user actions

It is very important that users are aware of the results of their actions. For example, in the Easy Snow app a student has the option to cancel a lesson, write a review, send an email to a coach, etc. When performing such an action, it is common for the user to wonder if their action has indeed been completed or needs to repeat it. For this reason, a corresponding update is displayed at the bottom right of the screen, as shown in Figures 49 and 50.

Easy Snow

Κράτηση Μαθήματα EN

Προφίλ

Μαθήματα

Αποσύνδεση

Σύμπτυξη

Μιλάω Ιταλικά και Ισπανικά	40+ χρόνια εμπειρίας
<input checked="" type="checkbox"/> Η πολιτική ακύρωσης είναι δωρεάν μέχρι <b>7 ημέρες</b> πριν το μάθημα	Μιλάω Ιταλικά και Ισπανικά
<b>Πληροφορίες συμμετεχόντων</b>	
∅ 1 άτομο ♂ Αρχάριος	
<b>Τηλέφωνο προπονητή</b>	
□ 309999999999	
<b>Επιλεγμένα μαθήματα</b>	
<b>Ακυρωμένο</b> Πέμπτη 26/06/2025 όλη μέρα (12:00-14:00)	40€
Σύνολο	40€
<b>Σημεία συνάντησης</b>	
<b>Ακυρωμένο</b> Πέμπτη 26/06/2025 12:00-14:00	Άνω σαλέ
Επικοινωνία	

Μιλάω Ιταλικά και Ισπανικά

Η πολιτική ακύρωσης είναι δωρεάν μέχρι **7 ημέρες** πριν το μάθημα

**Πληροφορίες συμμετεχόντων**

∅ 2 άτομα ♂ Αρχάριος

**Τηλέφωνο προπονητή**

□ 309999999999

**Επιλεγμένα μαθήματα**

**Ακυρωμένο** Πέμπτη 19/06/2025 08:00-12:00 80€

**Ακυρωμένο** Παρασκευή 20/06/2025 08:00-12:00 80€

Σύνολο 160€

**Σημεία συνάντησης**

**Ακυρωμένο** Πέμπτη 19/06/2025 08:00-12:00 Άνω σαλέ

To μήνυμα στάλθηκε

Image 49. Update a successful email send.

Easy Snow

Κράτηση Μαθήματα EN

Προφίλ

Μαθήματα

Αποσύνδεση

Σύμπτυξη

1 άτομο ♂ Μέτριος	
<b>Τηλέφωνο προπονητή</b>	
□ 309999999999	
<b>Επιλεγμένα μαθήματα</b>	
Δευτέρα 02/06/2025 όλη μέρα (09:00-15:00) 120€	
Σύνολο 120€	
<b>Σημεία συνάντησης</b>	
Δευτέρα 02/06/2025 09:00-15:00 Άνω σαλέ	
Κριτική	
Επικοινωνία	

Η κριτική σας υποβλήθηκε επιτυχώς

Image 50. Update of successful review submission.

### **Reset scrollbar position**

As we mentioned, building applications using React and React Router has some peculiarities compared to traditional applications. The main peculiarity is that the application is Single Page, i.e. all pages are displayed within one window. For this reason, we still need to somehow ensure that every time a new page is refreshed/loaded, the scrollbar is automatically moved to the top of the page. This is done automatically in traditional applications where the server sends a new html file at each page break but not in the SPAs. For this purpose, we use the `<ScrollRestoration/>` component of React Router (Devkar, 2025). This allows us to reset the scrollbar to the beginning during transitions and more sophisticated functions. Specifically, in the user's menu (image 48), if the user is at the bottom of the tab with the courses and visits his profile and then the tab with the courses again, then we have made the option to load the page in the same place as it was before visiting his profile. This gives the user the feeling that each of the menu tabs is a window that is simply hidden or displayed depending on where the user is, but not lost.

### **Protected pages**

A necessary element in all applications is the requirement to authenticate users when accessing protected routes. Of course, it is desirable that when users try to access a protected path, they are redirected to the application's login page and after entering they are automatically redirected to the page where they were located. Furthermore, to improve the user experience, it would be good if after logging in users decided to press the back button not to be redirected to the login page, since they would have already logged in. These functions have been developed in the Easy Snow app and are based on the official React Router code examples (Remix Run, e.g.).

### **Multi-criteria menu**

In applications with high interactivity, the user often needs to interact with menus. In fact, in modern applications, the menus are automatically updated with every change by the user without the user having to press a relevant search button. This gives the sense of immediacy and speed to the app and improves the user experience. For this reason, this technique has been used in the course selection menu which for convenience is repeated below in Figure 51.

Furthermore, in complex applications, the menus used are multi-criteria, i.e. users can choose from a variety of different filters. Specifically, in the Easy Snow app users can choose lessons based not only on the necessary restrictions such as sport, location, date and people but also with many others, such as whether the lesson is private or group, whether it is in the morning, at noon or if it is all day. Also, the courses can be sorted according to the coaches' reviews or their cost, either from higher to lower cost or vice versa. Also, the user can search for a specific coach based on his name. In fact, to further increase the feeling of immediacy, the search is done automatically when the user stops typing the name, without the user having to press another button.

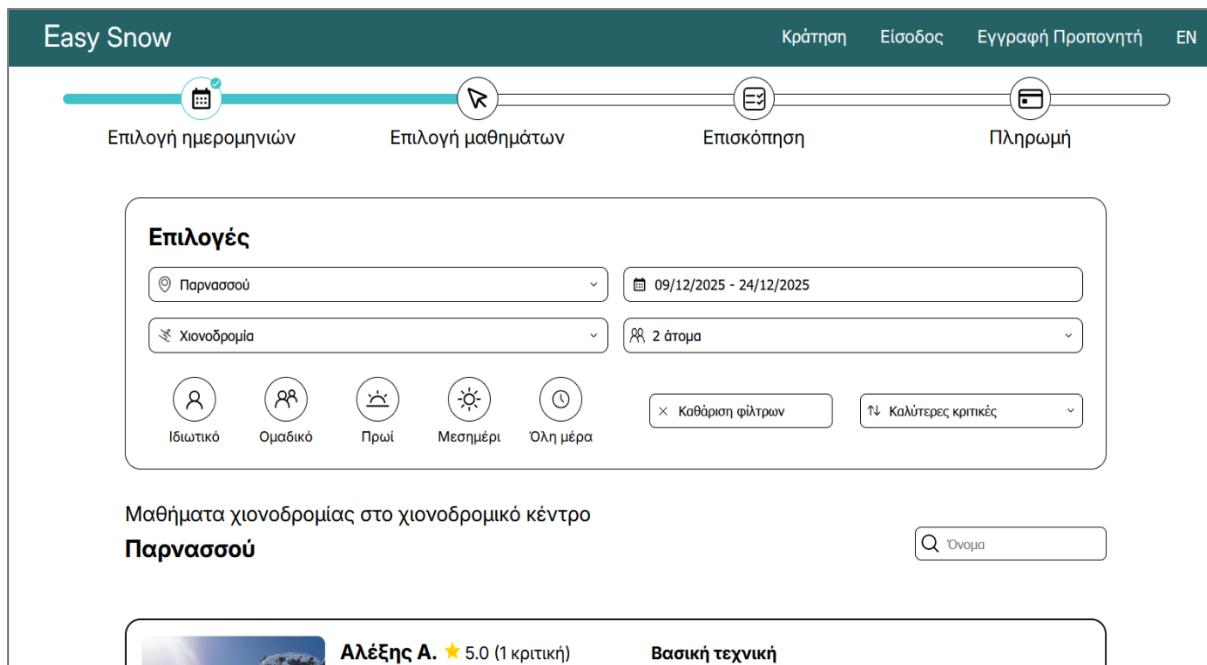


Image 51. Multi-criteria course selection menu.

### 6.2.3 Vite

As we mentioned at the beginning of this chapter, React uses jsx code files that extend traditional Javascript files in the sense that Javascript functions can return html code. However, jsx files cannot be read by the browser and therefore need to be converted into javascript files. Also, the generated files can be large and need to be sliced into smaller chunks, as we explained in section 6.2.2 **Possible difficulties in using React** regarding code splitting. These and many more are undertaken by a program known as a bundler.

#### Bundler Advantages

As mentioned above, bundler is useful because it converts jsx code into javascript code that the browser can understand and allows the code to be broken into smaller chunks. Of course, in addition to these, the bundler analyzes the dependencies in the code, identifies pieces of code that are not used, removes them from the final result, and optimizes the code to reduce its final size. Also, the bundler facilitates the development of the code as it allows the creation of a local server that simulates the client during the development of the code. In fact, because it makes sense that many changes occur during code development, it is especially important that the bundler quickly displays them to the client simulator (Eagles, 2022; Keshri, 2023).

#### Vite and Webpack comparison

The most prevalent bundlers are Vite and Webpack. Webpack is the oldest app created in 2012, while Vite is newer as it was launched in 2020. Webpack is considered more difficult to use and is noticeably slower than Vite, about 80% slower (Inyang, 2024). However, Webpack has the advantage of being more customizable than Vite, which makes it useful in specific applications with special needs. Therefore, based on the above comparison, we choose for the Easy Snow app to use Vite as it is faster.

## Vite configuration

As we mentioned, when developing the code Vite creates a separate local server that sends the jsx files to the client. At the same time, the customer using React Router loads the data they need from the server we have built. However, it is reminded that in order for the jsx code to be sent from our own server, it must first be bundled, which is a time-consuming process that cannot be done with every change during code development. For this reason, the server that automatically generates Vite is used to send the jsx code during the development of the application, and our server sends only the data that the website needs. This creates two problems. First, a page from Vite's server needs to make a request to our own (local) server, which is not allowed unless we specify that our server allows CORS (Cross-Origin Resource Sharing) requests. The second and main problem is that when using Vite's server, the URLs for retrieving the data will have to write the entire local address, while when the jsx code is sent from the end server that we will develop, the addresses will have to be relevant. For example, to retrieve data from the url "/api/mydata", the address needs to be "http://localhost:3000/api/mydata" when Vite's server is used and "/api/mydata" when our server is used. Of course, it is not possible to develop the entire application and in the end, before making the final bundled file, we can attempt to change all the URLs. These problems are solved by the options in the vite configuration file shown in Figure 52.

```
export default defineConfig({
  plugins: [react()],
  server: {
    proxy: {
      "/api": {
        target: "http://localhost:3000",
        changeOrigin: true,
        // secure:false,
      },
    },
  },
});
```

Image 52. Vite configuration.

First of all, we state that all requests that start with "/api" will be sent to "http://localhost:3000" so we can write all urls exactly as they will be when using our own server. Also, with the changeOrigin option, we solve the CORS problem we mentioned.

### 6.2.4 Internationalization (i18next)

The web applications are accessible from all over the world by people who speak different languages. For this reason, if one considers that Greece is a great winter tourist destination, one understands that the Easy Snow app should be accessible by tourists who may not know Greek. For this reason, it should also support foreign languages, such as English.

A very good library for the internationalization of an application is i18next (i18next, e.g.). This is because the library makes the internationalization of the application expandable to any number of foreign languages. Specifically, within the code files where we want some text to appear, we simply declare a key in its place. The translations are in separate json files, and the key combined

with the user's chosen language determines the specific text that users will see in the language of their choice. The reason why this architecture is scalable is because it allows the translation of each page to be in a separate file. Also, when the user initially visits the application, they do not need to load all translations of all pages, but only of each page at a time. In addition, even for the specific page that the user visits, translations of all languages are not loaded, but only those that the user needs. Therefore, only the minimum files are sent to the user at a time. That's why the app is fully scalable and can support dozens of languages as well as two. Specifically, for the Easy Snow application, Greek has been used as it concerns the ski resorts of Greece and English for tourists.

### 6.2.5 Charts ( Chart.js)

Coaches have access to statistics about their lessons, working hours, reviews, etc. For convenience, the charts are repeated below in Figure 53. Initially, the coach can see his working hours and how many hours he has taught private and group lessons for a specific month. Also, the coach can also see the income he had per month and per type of lesson, i.e. for private and group lessons separately. Furthermore, the coach can see the distribution of the grade of his reviews and see how many students rated him e.g. with 5 stars, how many with 4 and so on, as shown in figure 53. In fact, the Chart.js library has been used for the preparation of these charts, which offers a variety of charting tools (Chart.js, etc.).

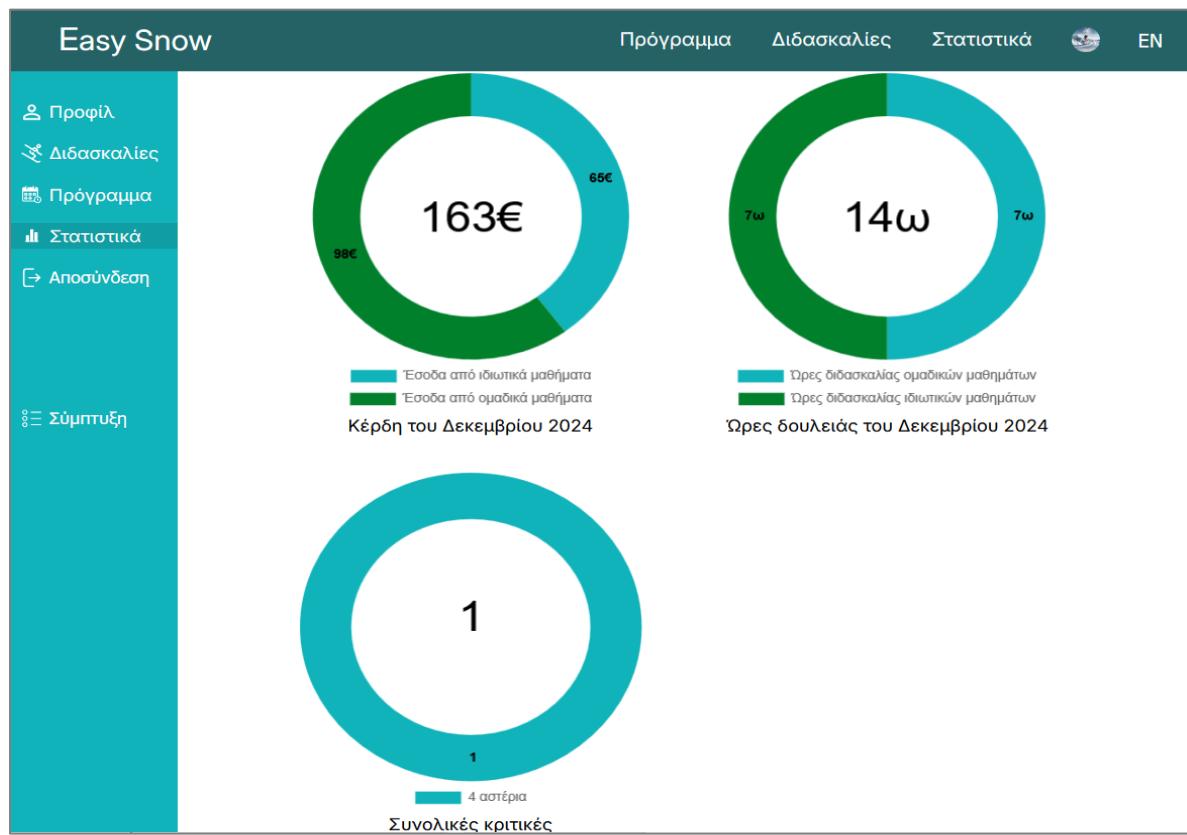


Image 53. Coach statistics.

## 6.2.6 CSS

CSS (Cascading Style Sheets) commands specify the presentation of the html elements. For example, they specify the color, size, layout of elements, font, etc. (MDN web docs, e.g.). In fact, as we mentioned, React integrates javascript and html code, but still needs to write CSS commands to format the pages. In fact, in the Easy Snow application, formatting has been done only with CSS commands, without any external library or ready-made code having been used. That's because CSS makes it possible to format the elements exactly the way we want, without having to rely on ready-made code with predefined features. Equally important, as we analyze below, is that CSS allows us to make our application adaptable to different device sizes, which is one of the non-functional requirements of Table 4.

### Responsiveness

As discussed in the PACT framework (section 4.5.2), the application needs to be accessible from devices with different screen sizes. For example, the coach may create his lessons from his computer or tablet and want to confirm something from his mobile phone when he is in the mountain. It is therefore highly understood that the Easy Snow app needs to be adapted to different screen sizes. To achieve this, media queries have been used, i.e. CSS commands that are applied depending on the size of the device.

Below are some examples of how the app appears on tablets and mobiles. In fact, a typical resolution size of a mobile screen is 360x800 pixels and for tablets 768x1024 pixels (BrowserStack, 2025).

- **Home screen**

The home screen for mobile is shown in figure 54. We notice that the layout is vertical to adapt to the size of the mobile. Of course, on this page the presentation for tablets is similar to the presentation for computers shown in Figure 24.

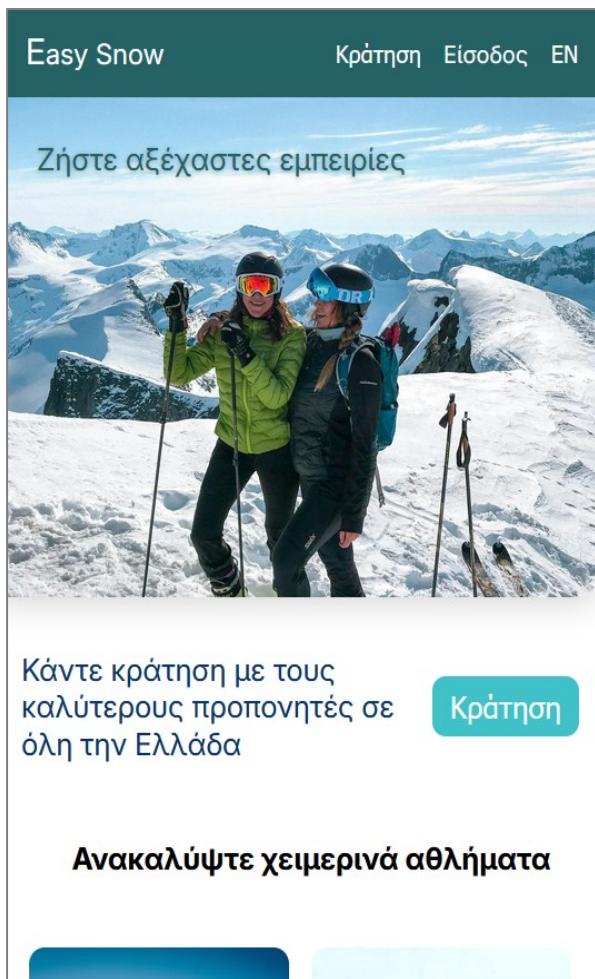


Image 54. Mobile Home Screen

- **Reviews**

Figure 55 shows user reviews on mobile. In fact, a vertical layout has been chosen again so that reviews are visible on a screen with a limited range. Figure 56 shows the corresponding page on a tablet, where the reviews have a 2x2 layout as the tablet is longer.

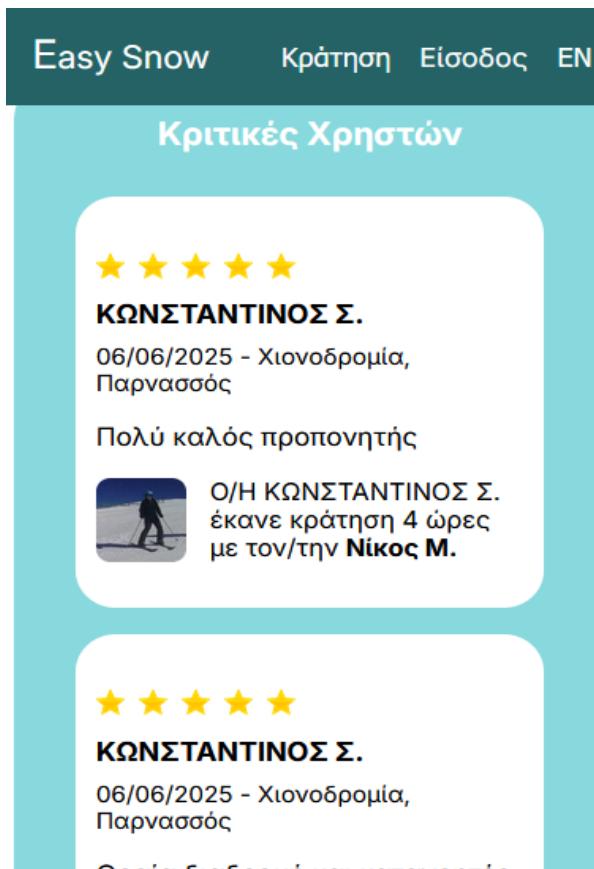


Image 55. Mobile user reviews.

- **Signup**

The mobile registration page is shown in Figure 57. Specifically, both vertical and horizontal layout have been chosen. The larger components are in a vertical arrangement because they need a long length. Of course, some components that have a shorter length are in a horizontal arrangement. Specifically, the code and the code confirmation are on the same line and respectively the country code of the mobile phone along with the phone number are on the same line because they have a conceptual connection. Also, the registration page on a tablet is the same as on a computer (figure 27).

## Κριτικές Χρηστών



Image 56. User reviews on tablets.

- **Choice of courses**

Images 58 and 59 show the course selection menu on the tablet and mobile respectively. In particular, in Figure 58 we notice that a mixed arrangement is used. The basic options have a 2x2 layout and the additional filters have a horizontal layout. Also, the lessons are placed in pairs.

Figure 59 shows the corresponding page on mobile. The basic menu options for location, sport, people, and dates are arranged vertically. Also, the additional filters have a horizontal arrangement because they require little space. Finally, the courses are in a vertical layout, as a mobile screen only holds one lesson.

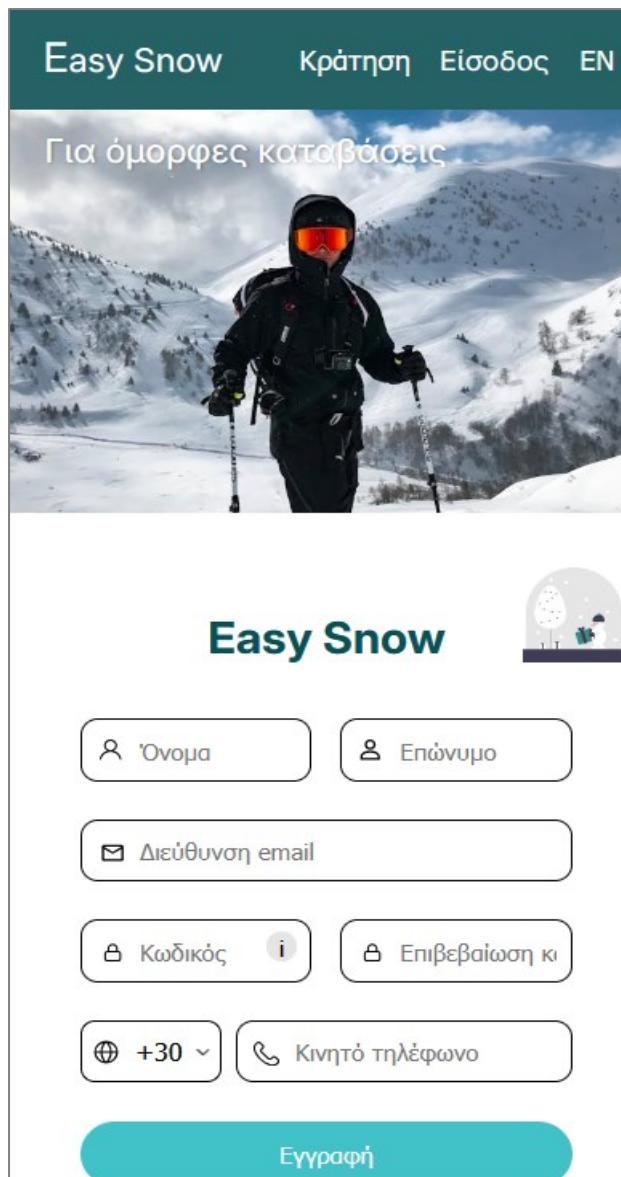


Image 57. Mobile registration page.

Easy Snow

Κράτηση Είσοδος Εγγραφή Προπονητή

### Επιλογές

Παρνασσού 09/12/2025 - 31/12/2025

Χιονοδρομία 3 άτομα

Ιδιωτικό Ομαδικό Πρωί Μεσημέρι Όλη μέρα

Καθάριση φίλτρων  ↑↓ Καλύτερες κριτικές

Μαθήματα χιονοδρομίας στο χιονοδρομικό κέντρο  
Παρνασσού

Όνομα



Αλέξης Α. ★ 5.0 (1 κριτική)  
Βασική τεχνική



Νίκος Μ. ★ 5.0 (2 κριτικές)  
Traverse

Image 58. Choice of courses on tablets.

Easy Snow

Κράτηση Είσοδος EN

### Επιλογές

Παρνασσού 09/12/2025 - 31/12/2025

Χιονοδρομία 3 άτομα

Ιδιωτικό Ομαδικό Πρωί

Μεσημέρι Όλη μέρα

Καθάριση φίλτρων  ↑↓ Καλύτερες κριτικές

Μαθήματα χιονοδρομίας στο χιονοδρομικό κέντρο Παρνασσού

Όνομα



Image 59. Choice of courses on mobile.

- **Checkout page**

The mobile payment page is shown in Figure 60. In fact, the layout that has been chosen is again vertical as the mobile has a limited screen width. For tablet devices, the layout is the same as a computer (Figure 43).

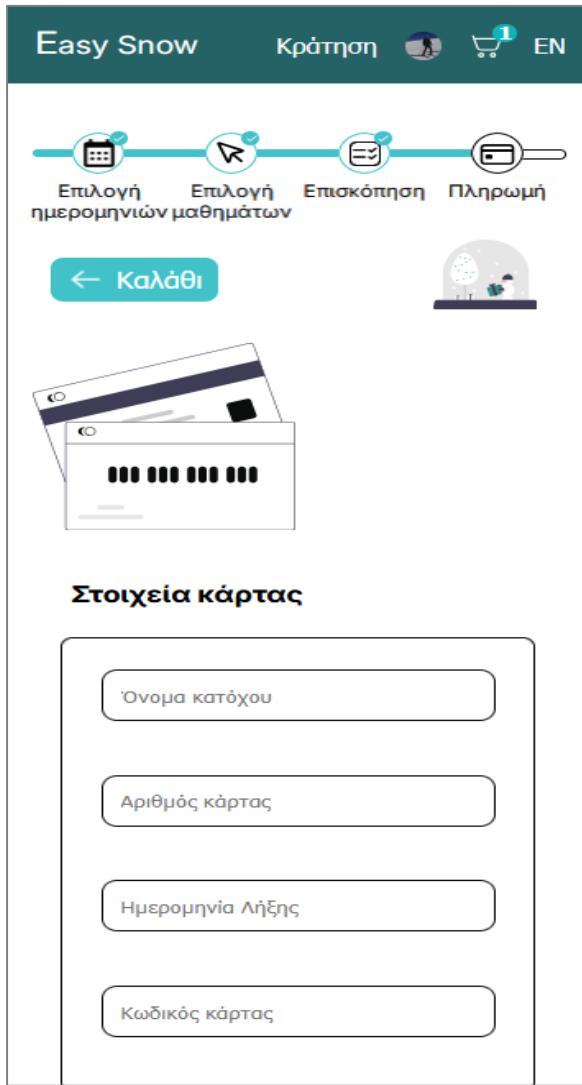


Image 60. Mobile payment page.

- **Creating new teaching**

Figure 61 shows the page of creating a new teaching by a coach on the tablet. The layout that has been used is two columns as there is sufficient space. Of course, a vertical layout with a single column is used on a mobile phone, as shown in Figure 62. Also on mobile, the options in the header (program, tutorials and statistics) are accessible if the user taps on their profile picture, as space is limited.

Easy Snow

Πρόγραμμα Διδασκαλίες Στατιστικά EN

Δημιουργία διδασκαλίας

<b>Χιονοδρομικό κέντρο</b>	<b>Αθλητισμός</b>
Χιονοδρομικό κέντρο	Δραστηριότητα
<b>Ημερομηνία έναρξης</b>	<b>Ημερομηνία λήξης</b>
Επιλογή ημερομηνίας	Επιλογή ημερομηνίας
<b>Ημέρες μαθημάτων</b>	<b>Ώρα έναρξης - Ήρα λήξης</b>
Επιλογή ημέρων	Π.χ. 8:00 - Π.χ. 12:00
<b>Το μάθημα διαρκεί όλη μέρα</b>	<b>Τύπος μαθήματος</b>
Nαι	Όχι
Iδιωτικό	Ομαδικό
<b>Μέγιστο πλήθος συμμετεχόντων</b>	<b>Σημείο συνάντησης</b>
Πλήθος σπόλων	Σημείο συνάντησης

Image 61. Create a coach teaching on a tablet.

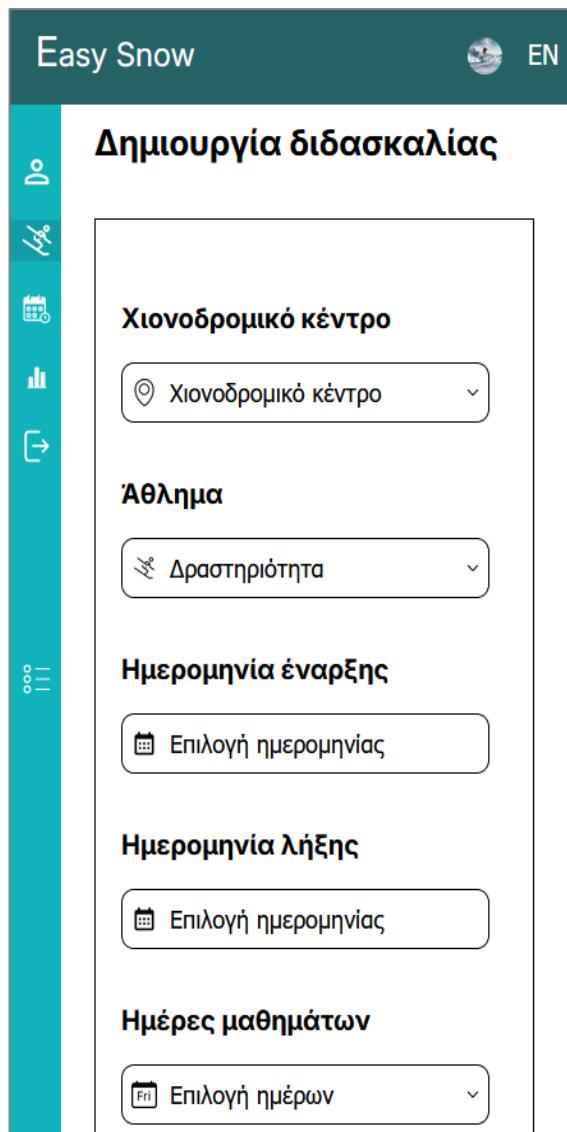


Image 62. Create a coach teaching on mobile.

- **Coach Program**

Figure 63 shows the coach's schedule page on a tablet. In fact, we notice that for the subsections of each course, i.e. the course elements, the meeting point and the students' data, a vertical layout has been used. Of course, within the subsection for the lesson elements, the layout is 3x2 as two elements fit on the same line. However, in the case of the mobile phone, the device used is vertical, as usual, due to limited space and is shown in Figure 64.

Easy Snow

Πρόγραμμα Διδασκαλίες Στατιστικά EN

Προφίλ Διδασκαλίες Πρόγραμμα Στατιστικά Αποσύνδεση

Σύμπτυξη

## Πρόγραμμα

24/12/2025

### Μαθήματα

**Στοιχεία μαθήματος**

- Χιονοδρομικό Παρνασσού
- Αθλημα Χιονοδρομία
- Ημερομηνία Τετάρτη 24/12/2025
- Συμμετέχοντες 1
- Ώρα 08:30-15:00
- Τύπος μαθήματος Ιδιωτικό

**Σημείο συνάντησης**

Αφετηρία 2

**Στοιχεία μαθητών**

Γιώργος Γεωργίου 306951231212

Image 63. Coach program page on tablet.

Easy Snow

EN

Προφίλ Διδασκαλίες Πρόγραμμα Στατιστικά Αποσύνδεση

### Μαθήματα

**Στοιχεία μαθήματος**

- Χιονοδρομικό Παρνασσού
- Αθλημα Χιονοδρομία
- Ημερομηνία Τετάρτη 24/12/2025
- Συμμετέχοντες 1
- Ώρα 08:30-15:00
- Τύπος μαθήματος Ιδιωτικό

**Σημείο συνάντησης**

Αφετηρία 2

**Στοιχεία μαθητών**

Γιώργος Γεωργίου

Image 64. Coach program page on mobile.

## 6.3 Server-side technologies

Server means the program responsible for sending the code and page data requested by the user. The server also receives various requests from users and takes appropriate actions, which may involve modifying a database. The technologies used on the server side are broken down below.

### 6.3.1 Node.js

Node.js is an open-source, free JavaScript execution environment designed for the development of high-performance web applications. It is based on Google's V8 engine and allows JavaScript to be executed on the server-side, extending its use beyond the browser. With an event-driven architecture, Node.js is ideal for applications that require simultaneous service of multiple requests, such as APIs, web services, and real-time applications (Nodejs, e.g.).

In fact, it is especially important that the Node.js ecosystem is supported by npm (Node Package Manager), which provides access to one of the largest collections of open source software worldwide. The project is supported by the Node.js Foundation and actively maintained by a broad community of developers, while offering regular updates, LTS (Long Term Support) releases, and extensive documentation.

Furthermore, the platform works on multiple operating systems (Windows, Linux, macOS) and is fully scalable, allowing the integration of third-party libraries.

For all of the above reasons, the server has been deployed in a Node.js environment.

### 6.3.2 Express.js

The route to the server has been done with Express. Express is a flexible web framework for Node.js, designed to make it easy to build web applications and APIs. In fact, it provides a variety of functions for creating applications while allowing rapid development with less code and maximum scalability (Express, e.g.).

Furthermore, Express is open source software and is a key component of the Node.js ecosystem, offering advanced routing, HTTP request management, middleware, and view engines. Of course, express viewers are used in the traditional application development model and not in Single Page Applications built with React, so they were not used in this application.

Additionally, Express's architecture is middleware-based, allowing for easy addition of features to the app's core. Therefore, it is fully scalable and can be customized according to the needs of each project. That's why it's widely used in the industry to build RESTful APIs and backend services.

In conclusion, Express has extensive documentation, is supported by an active community, and combined with Node.js, offers an efficient platform for creating modern, fast, and scalable online applications.

### 6.3.3 Data validation

The validation of the data entered by the user is an essential element for the security, stability and proper functioning of any web application. Often, the data entered may be incorrect, either due to misinterpretation of the instructions by the user, or due to malicious intent. For this reason, it is necessary to carry out checks at multiple stages of the import process.

The first check occurs in the on blur event, which is when the user moves away from a form input field. This check allows for instant feedback, helping the user correct potential errors without having to complete the form. In this way, the interaction becomes more friendly and interactive. In fact, this check is very important in the Easy Snow application during the registration process, where the user is asked to enter an email address, and if it is already occupied then the user is automatically informed when he changes field without having to fill out the entire form first.

The second check takes place during the submission of the form, where the validity of all fields on the client-side is checked as a whole. This check ensures that a user who does not try to trick the application will send valid data to the server.

Finally, the third and most important check is carried out on the server-side. Even if previous audits show that the user is entering valid data, auditing on the server is necessary to troubleshoot malware and protect the application's database and operations, as a malicious user can bypass previous audits (SurveyJS, e.g.).

### 6.3.4 EmailJS

The Easy Snow app enables users to ask questions to coaches through the app. For this purpose, the EmailJS application has been used, which enables e-mails to be sent programmatically (EmailJS, e.g.). In fact, the recipient's address is defined to be that of a coach and the address to which the coach will respond is the address of the user who asked the question.

### 6.3.5 Safety (Helmet)

To enhance the security of the application, the Helmet library is used, which is a library for Node.js applications that enhances security by adding special HTTP headers – that is, small pieces of information that are sent along with each server response to the browser. These headers help prevent common cyberattacks.

One of the most important features of Helmet is the Content Security Policy (CSP). CSP allows the developer to determine from which sources content is allowed to be loaded (e.g., scripts, images, styles). If a hacker tries to inject malicious JavaScript code into a page (known as XSS – Cross Site Scripting), Helmet sets an appropriate CSP that will prevent it, as this code does not come from the allowed sources (Jacob, 2023; Talmi, 2023).

Another important header is X-Frame-Options, which prevents the page from appearing inside the iframe of another web page (Zanini, 2023). This protects against clickjacking, a technique where the user is tricked into "clicking" on something dangerous without knowing it.

Finally, Strict-Transport-Security (HSTS) obliges browsers to connect only via HTTPS. This prevents data from being tracked on insecure connections.

## 6.4 Database

The Easy Snow application manages a variety of data such as users, bookings, courses, teachings, reviews and so on. To store this data, a database needs to be created.

### 6.4.1 Entity-relationship diagram

An Entity-Relationship Diagram is a graphical tool used to represent the structure of a relational database. It depicts the basic entities of a system, their attributes and the relationships between them. Entities represent real-world objects, such as "Customer" or "Order", while relationships show how they are connected to each other, e.g., "contains", "buys", etc. (Creately, 2024).

In fact, ERD is an essential tool at the stage of designing a relational database, as it helps in rational design and understanding of the logical structure of data before implementation. It allows developers, analysts, and stakeholders to collaborate on a common representation of system requirements.

In addition, ERD helps prevent design errors, normalize data, and ensure relationship integrity. In summary, it provides a clear, documented picture of the information that will be managed by the system, making future maintenance and expansion of the database easier and more secure.

The correlation entity diagram of the Easy Snow app is shown in Figure 65 and has been developed through the ERD Maker tool (ERD Maker, 2021). Specifically, we notice that the user of the application specializes in a coach and a student. Coaches can also create lessons that contain lessons. For example, a lesson can be about skiing every Monday 8:00-10:00 and Tuesday 11:00-13:00 for the whole winter season. The lesson of a lecture concerns a specific date. In addition, coaches set meeting points, which may contain a verbal description and an image of a map with the point where their lessons will begin. Also, students can book lessons or add them to their basket. In addition, students can write reviews of their lessons and evaluate coaches. Finally, there is also the corresponding entity for making the payments.

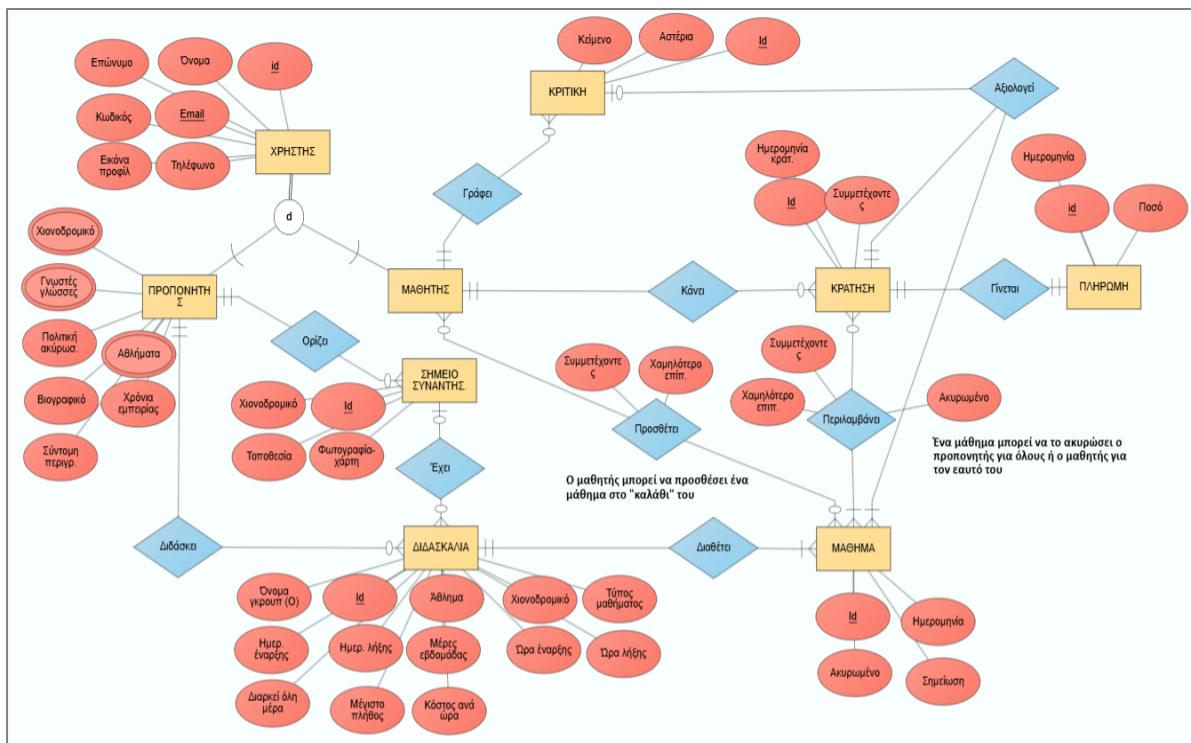


Image 65. Entity-correlation diagram of the Easy Snow application.

#### 6.4.2 PostgreSQL

PostgreSQL has been used as a database, which offers a powerful, open-source relational database management system (RDBMS), which is actively supported by a global community of developers. In particular, PostgreSQL is characterized by reliability, stability, and scalability, which makes it ideal for commercial and academic applications. In fact, according to research (Stack Overflow, 2024) PostgreSQL is the most popular database.

Furthermore, PostgreSQL supports indexes (B-tree) for efficient data search and processing, while providing full extensibility with custom formulas, functions, and extensions (Bantia, 2023).

Moreover, PostgreSQL applications cover a wide range of areas, such as business information systems (OLTP), data warehouses, analytics, GIS systems, web applications, scientific research and IoT environments.

In summary, PostgreSQL combines high performance, scalability, active community, and support for modern technology requirements, making it a stable and reliable solution for applications like Easy Snow looking for secure, flexible, and scalable data management.

#### 6.4.3 Indexes

Indexes are a critical mechanism in a database management system, as they significantly optimize the performance of search queries. After all, they are used to retrieve data from tables faster, without the need for a full scan of records. In particular, indexes act as data structures that allow direct access to specific values based on one or more fields.

Therefore, the use of indexes drastically reduces the response time to SELECT commands, particularly when dealing with large databases. At the same time, it enhances efficiency in functions that include classification, comparison or joins. For the above reasons in the Easy Snow application, in addition to the indexes in the primary key that are automatically generated (e.g. PostgreSQL), indexes have been created in the other fields of the tables that are often needed in SQL queries.

In fact, although indexes consume additional storage space and may burden import or update functions (INSERT/UPDATE), the benefit in search and execution speed is very significant. Therefore, the use of indexes is particularly useful for the Easy Snow application.

#### 6.4.4 User password protection

The secure storage of passwords is a fundamental necessity of the security of information systems and is at the same time a basic non-functional requirement of the Easy now application (Table 4). In fact, it is known that passwords should never be stored in plain text format, as in the event of a database breach, those who access the database can see all user passwords (Ilot, 2023). This, combined with the fact that most users use the same passwords in various applications that can be very important, such as bank accounts, makes it clear how necessary it is to protect users' passwords.

The basic approach to password protection involves the use of one-way encryption (hashing) techniques, which convert the code into an irreversible value. During the authentication process, the stored hash is compared to that generated by the imported code, without the need to save the original code.

Another important element is the addition of a unique and random "salt" for each code before hashing. Salt ensures that even if two users have the same code, their hash values will be different, making the use of pre-built tables to predict the code inefficient. At the same time, the use of powerful algorithms such as bcrypt, used in the Easy Snow app, adds several rounds to the hashing process, making it particularly computationally demanding for brute-force attacks.

Specifically, it is estimated that a crafty person using modern technological means, would need about 27,000 years to crack a 9-character code containing numbers, lowercase and uppercase letters, and symbols (White, 2023).

### 6.5 Implementation time required

Section 4 in Table 3 identified the functional requirements of the application. In fact, for each requirement, an estimate of the expected time of its implementation was given. Having completed the implementation of the application's functions, we are now given the opportunity to reflect on whether these estimates were accurate.

In general, the estimates did not turn out to be accurate as it took much longer to implement the functions. Specifically, requirement 3 for the promotion of available coaches according to criteria (location, dates, sport, etc.) took about 15 days to implement, i.e. three times the expected time. This is due to the fact that the multi-criteria menu shown in Figure 29 has been created. This proved to be demanding and time-consuming as every time the user selects a filter, the

corresponding lessons are automatically shown to him without having to press another button. Furthermore, with each filter change, the url automatically changes so that the user can, for example, save the page together with the filters he has selected. Furthermore, all the individual filters are eventually translated into parameters of a SQL query, which needed several auxiliary tables to implement and had a complex structure.

In addition, the requirement to be able to define the language of the application took about 15 days to implement, which is twice as long as expected. This is because in order to implement it, familiarization with the i18next library had to be introduced, which took about a week. Then a separate folder had to be created for each language. In this case there are two languages in the application, Greek and English, so two folders were created. Then, in each folder for each page of the application, a json file containing the key-translation pairs had to be created. Then, on each page, all the texts had to be replaced with the corresponding keys, a process that required about another week.

## 6.6 Conclusions

In this chapter, the code for the Easy Snow app was developed. In fact, React, which is an extremely widespread library for the development of modern Single Page Application (SPA) applications, was utilized. Also, the history was managed with the React Router library, also a very popular library. Furthermore, the server was developed in a Node.js environment using the Express.js library. PostgreSQL was also used for the database, which is one of the most widely used open source database systems. Finally, a comparison of the estimated and actual deployment times of the application's operational requirements was made and it was explained that the actual times were much longer than expected.

Therefore, a comprehensive evaluation of the functions of the application can now be made in the next chapter with both expert and user evaluation methods.



# **7 Evaluation**

The evaluation of the application is a basic process of the human-centered design (section 2.3.3) and the asteroid development model (section 2.3.4) that has been used in the Easy Snow application. In fact, evaluation methods are divided into evaluation methods by experts, i.e. people who know design principles and rules, from users, i.e. people who use the application, and from analytical evaluation methods based on the hierarchical analysis of tasks, which has been done in section 4.4 (Avouris et al. 2018).

However, before the assessment begins, it is important to mention that all the operational requirements set out in Table 3 have been fully implemented. After all, the design was also done in such a way as to fully meet the functional requirements of the application. Therefore, we can now evaluate the implementation in its final stage without there being any points to be developed. Of course, as far as the non-functional requirement of Table 4 for data security is concerned, this was implemented only in user passwords, as passwords are extremely important to be protected even if someone skillfully gains access to the database, while the rest of the users' data is not considered so sensitive.

## **7.1 Methods used**

The evaluation of the Easy Snow app was done in several stages. Initially, the design of the application was evaluated using the Hink Aloud Protocol method (section 5.4.1) in order to immediately identify potential usability issues and correct them before implementation. Subsequently, after implementation, both expert and user evaluation methods were used. Specifically, for the evaluation with experts, a Heuristic evaluation and a Simplified Cognitive Walk were performed. For the user assessment, the Talking Subject Protocol, semi-structured interviews and the questionnaire method were used.

## **7.2 Expert Evaluation**

Expert evaluation methods are simple in the sense that they are carried out without the involvement of many users and do not take much time to prepare. Therefore, the expert evaluation will be done prior to the evaluation with users to correct any usability issues that may be identified. The app will then be evaluated by the end users, which takes time but is necessary to determine if the app is indeed considered easy to use by the people for whom it was created.

### **7.2.1 Heuristic evaluation**

The Heuristic evaluation is based on 10 basic rules that have emerged from the experience of experts and contribute to the usability of the application (Nielsen, 1993; Avouris et al. 2018). In fact, it is a formative evaluation, i.e. it focuses on specific points and aims to highlight errors.

In fact, the Heuristic evaluation was made by the author, who tried to be objective but due to his familiarity with the application, he may have a positive predisposition towards it and the evaluation may be favorable. This is another reason why there will be an evaluation with the end users.

Below are the rules of Heuristic evaluation and how the Easy Snow app responds to them.

#### **1. Update on the status of the system**

Users must know where they are on the website and at the same time what the results of their actions are. This is achieved in the Easy Snow application, as in the navigation menu of both the coach and the student (images 41 and 42 respectively), the tab they are in each time has a different color from the others. Also, when users take an action, e.g. cancel a lesson or send an email to a coach, they are informed about the successful or unsuccessful implementation of their action (figure 49).

#### **2. System-to-real world mapping**

The language of the app and the symbols must correspond to reality and what users are already familiar with. This happens on all pages of the Easy Snow app as the words are familiar and understandable. Also, all the symbols are easy to understand, as is the case for example in the course selection menu in figure 29, where for the date there is a calendar, for the sport a man skiing, for the noon a sun, for the classification of the lessons two arrows up and down and so on.

#### **3. Freedom and control by the user**

There must be clean outputs from each page as well as the ability to go back. This is true because in the application, by clicking on the Easy Snow logo located in the header, the user is returned to the homepage. Also, the student when viewing courses in the cart can click back to return to the course selection page and add more (figure 33).

#### **4. Consistency and continuity and standard use**

The navigation model and page structure must be consistent throughout the application. Indeed, in the Easy Snow app the header and footer are in fixed positions throughout the application. Similarly, the logo is always located in the header at the top left. In addition, there is consistency in colors, icons, application language and font.

#### **5. Avoiding mistakes**

The application must recognize errors and prevent them from occurring. As we explained in section 6.3.3, when submitting data to a form, three different control phases occur. The first occurs when the user changes a field and is automatically informed if the field it was in has been filled in correctly. The second phase occurs when all fields have been filled in and the user submits the form. In fact, Figure 66 shows the error messages to the user when they have filled in the registration fields incorrectly. Finally, the third control phase occurs on the server and ensures the validity and correctness of the data handled by the server.

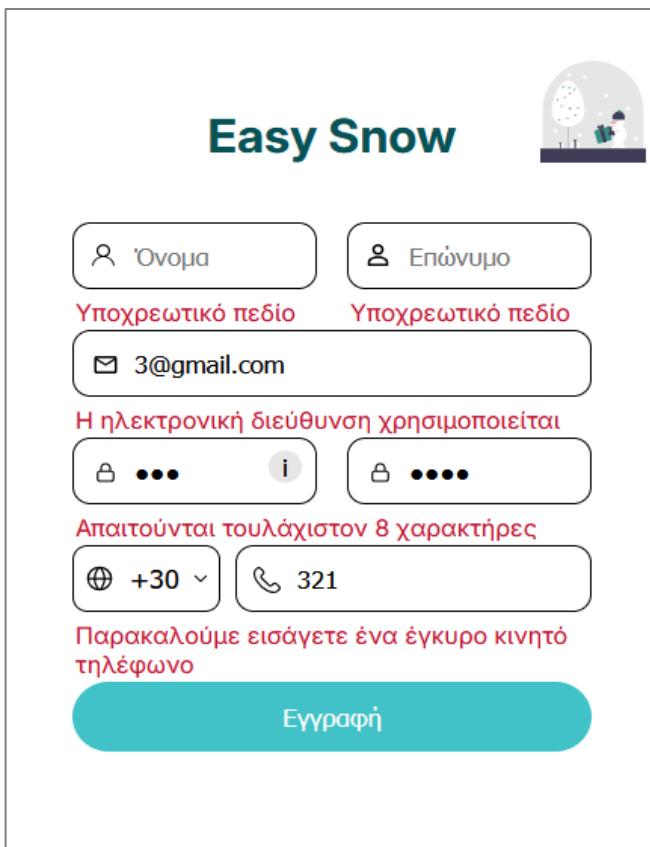


Image 66. Informing user about incorrect input fields.

## 6. Recognition instead of revocation

The links on the page must have correct names so that users can immediately and quickly figure out which link they want to choose without having to remember things from their previous interaction with the app. This is achieved in the Easy Snow app using understandable descriptions, such as "Status", "Courses", "Schedule", "Profile", "Statistics", etc.

## 7. Adaptability and efficiency of use

The use of advanced search techniques, saving pages, and customizing the page to users must be supported. Indeed in the course selection menu in image 29, we notice that the user can search for courses with varied and different filters at the same time. For example, it can search for courses based on the time of the lesson, its type or even the name of the coach. In addition, as mentioned above, the React Router library was used for the client-side routing, which allows the history to be changed and, by extension, to save different pages. Finally, in the navigation menus of both the coach and the student (images 41 and 42 respectively), the navigation bar can be minimized and not the letters but only the icons can be seen, which may be preferable for users who are already familiar with the application and navigation options.

## 8. Elegant and minimalist design

There should be no unnecessary information that tires users and the design should be clean and simple. This has been achieved as white space, soft and uniform colors and the inclusion of only the necessary information at any given time is the key element of the design of the Easy Snow app. Such a typical example is the homepage repeated in Figure 67.

## **9. Recognition and recovery from mistakes**

The application must recognize the user's errors and explain to them in an understandable way how to fix them. This has been achieved by checks on the data entered by the user and the use of understandable error messages. For example, in figure 66 during registration, the user is informed that he must fill in the required fields, that the email address he has entered is in use, that the code must be at least 8 characters long and that he has not entered his phone correctly.

## **10. Help**

The user should be offered help when actions are relatively complex and where he or she may be wrong. In fact, a typical example of assistance in the Easy Snow application is during registration where when filling in the code there is an i symbol (from the information) that provides information on what the user's password should include, as shown in figure 68.

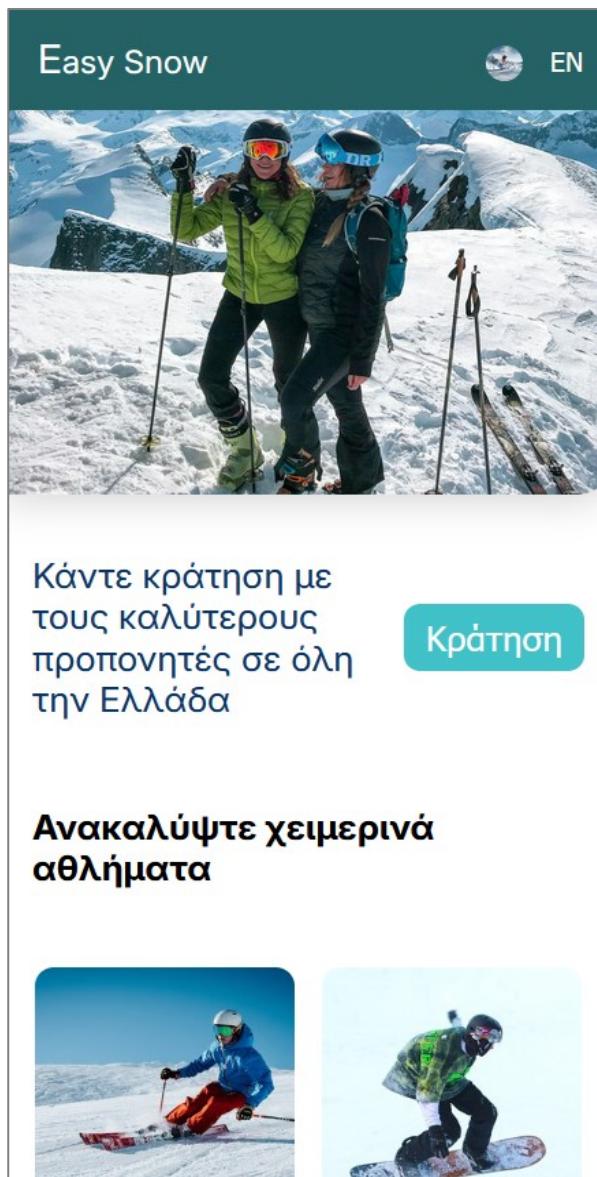
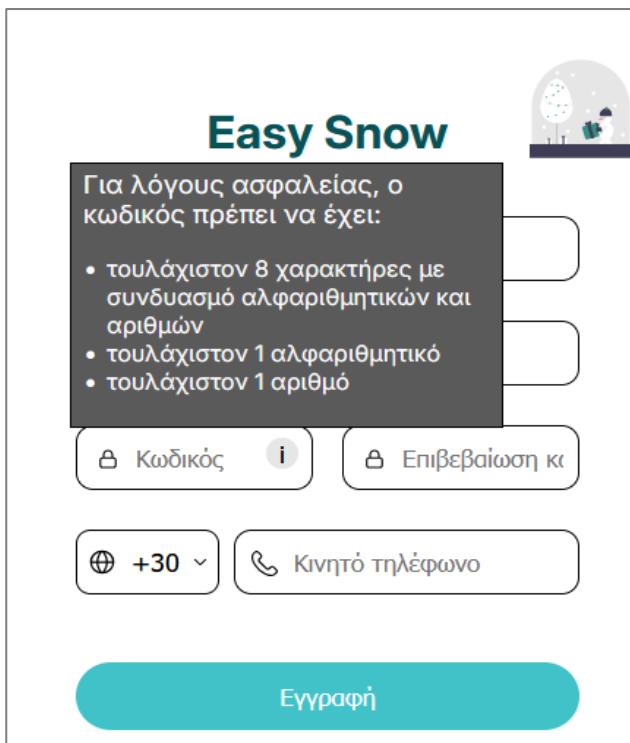


Image 67. White space and soft colors in the Easy Snow app.



*Image 68. Explanation of necessary code characteristics.*

### 7.2.2 Cognitive Jogthrough

Cognitive Walkthrough (Lewis et al. 1990) concerns the correlation of the user's goals with the functions of the system, in the context of specific usage scenarios. This method aims to evaluate the user-friendliness of the application by putting ourselves in the user's shoes and implementing typical scenarios of using the application to understand the thoughts that the user has at each step and to confirm that there is a logical sequence of the required steps and that these are fully understood. However, in the context of the evaluation of the Easy Snow application, Cognitive Jogthrough (Rowley & Rhoades, 1992) was used. In line with this, at each step we ask the following two questions:

- Do I know what I should do in this step?
- If I do the right thing, will I know that I have done the right thing and that I am making progress towards my goal?

In fact, the Cognitive Jogthrough for the student was applied to the course planning, as identified during the hierarchical task analysis (section 4.4), as it is the most complex and time-consuming process. Furthermore, for the coach, the process of supervising his program for a specific day was chosen, as it is a process that a coach will carry out frequently. Tables 5 and 6 show the results of the process.

However, the Cognitive Jogthrough was done by the author, who tried to be objective but due to his familiarity with the application, he may have a positive predisposition towards it and the evaluation may be favorable. This is another reason that there will be an evaluation with the end users in the future.

*Table 5. Results of Cognitive Jogthrough for Lesson Planning.*

n/a	Do I know what to do?	Since I did the right thing, do I understand that I did the right thing and am making progress?
Step 0: Start the process	Yes, I need to select a booking from either the header or the homepage.	Yes, as it opens a new window for selecting the course parameters.
Step 1: Select Course Parameters	Yes, there is a short explanatory text and options for location, sport, participants and dates.	Yes, as the Next button changes color to bright blue.
Step 2: Choosing a coach	Yes, as I am redirected to a new page with the available coaches and each coach has the option to "View Courses".	Yes, because I see a new window opening with the available lessons of the coach.
Step 3: Choose Courses	Yes, because I see the available courses and can choose them.	Yes, because the selected courses are displayed in blue and at the bottom there is information about the selected courses as well as the possibility of overview and payment.
Step 4: Visitation	Yes, as the progress bar shows that I am in the overview step and can see the courses I have chosen. Also at the bottom there is the option "Book".	Yes, as I'm taken to a new page and the progress bar fills up and now shows that I'm in the last step which is payment.
Step 5: Payment	Yes, as the progress bar shows that I am in the "Payment" step and I need to fill in my card details and press "Payment".	Yes, as I am informed by a message that the payment has been made.

*Table 6. Results of Cognitive Jogthrough for the supervision of a coach program.*

n/a	Do I know what to do?	Since I did the right thing, do I understand that I did the right thing and am making progress?
Step 0: Start the process	Yes, I have to click on the page titled "Program".	Yes, because a new page opens with the title program.

n/a	Do I know what to do?	Since I did the right thing, do I understand that I did the right thing and am making progress?
Step 1: Select a date	Yes, I have to click on the calendar and choose the date I'm interested in.	Yes, because the scheduled lessons of that day are displayed.
Step 2: Supervise Courses	Yes, I need to check for the courses I'm interested in, the meeting point, the time, the participants, and the sport.	Yes, because I oversaw my schedule for a specific date.

We notice that the answer to all questions is yes, suggesting proper application design, and that the steps that users need to follow in each scenario are fully understood.

### 7.3 User evaluation

Even if the expert evaluation did not highlight any usability issues, it needs to be evaluated by users because some elements of the application may not be understood by end users. That's because end users typically have different characteristics than experts and may be less familiar with technology. In fact, another important reason we conduct a user assessment is to assess whether the non-functional requirements of Table 4 (Non-functional Requirements Table) have been met. Specifically, we aim to find out how easy and intuitive the app is to use and whether the design is simple and tasteful. Finally, the evaluation with users is necessary because the evaluation by experts has been done by the author and creator of the Easy Snow application, who may have a positive predisposition towards the application and the evaluation he has made may be favorable. For the above reasons, we will carry out an evaluation with the end users and in particular we will use the Talking Subject Protocol, semi-structured interviews and the questionnaire method.

#### 7.3.1 Thinking aloud Protocol

In this assessment method, users are asked to take representative actions with the application and externalize their thoughts (Lewis, 1982). In fact, before the evaluation started, we told users that their thoughts can be anything related to the app and the process they are doing, e.g. it can be something they don't know how to do, something that confuses them or something they like. Also, Table 7 shows the characteristics of the users who participated in the evaluation and Table 8 shows the scenarios they were asked to carry out.

*Table 7. Characteristics of users who participated in the assessment using the method of interviews and the Thinking Aloud Protocol.*

User	Features

---

Spyros S.	He is 28 years old. He has been involved in skiing for 10 years.
Thanos M.	He is 40 years old. He has been involved in snowboarding for about 15 years and has taught several of his friends.
Angelos M.	He is 23 years old. He has taken ski lessons in the past.
Philippos S.	He is 23 years old. He is interested in taking his first ski lesson.

---

*Table 8. Scenarios performed by users during the assessment using the method of interviews and the Talking Subject Protocol.*

---

n/a	Scenario
1	You want to register for the app as a student. Pretend you don't know Greek and change the language of the app to English.
2	Do you want to hold a ski lesson on Parnassos this coming Christmas? Choose a coach who meets your needs.
3	You want to ask the coach with whom you scheduled a lesson for clarifications on what the meeting point is. Send them an email through the app.
4	Suppose you are Michalis Zervas, a ski coach on Parnassos. Create a new snowboard tutorial that takes place every day for the entire winter season.
5	Suppose you are Michalis Zervas, a ski coach on Parnassos. Cancel from the lessons you created what happens during the Christmas holidays, because you forgot that then you don't want to take a class.

---

In fact, during the above evaluation process, users reported that they liked the app's soft colors, the simple design with enough white space, and the absence of unnecessary information. However, two respondents indicated that it would be desirable when selecting the dates in Figure 28 to show which dates are available and which are not. Therefore, this is an element that would be desirable to be added to the application, which will be done later. Furthermore, the findings of the evaluation with users are then presented in aggregate in Table 9.

### 7.3.2 Interviews

Interviews are one of the most widely used methods of evaluation by end users as they serve to record the opinions of the app's users. Of course, the evaluation with interviews aims both to assess whether the non-functional requirements of the application for easy and intuitive design have been met, as well as to highlight any usability problems. In fact, semi-structured interviews

were used, i.e. the researcher has prepared some questions, but the order in which they are asked may change. Also, the researcher can ask additional questions based on the flow of the conversation.

In the process of evaluating the Easy Snow application with users, the Talking Subject method was initially applied, which was analyzed in the previous section. Subsequently, interviews were conducted with the same users, whose characteristics are recorded in Table 7. However, regarding the number of participants, it is noteworthy that the ideal number of users for evaluating a product is considered only three to five users at a time. This is because it is estimated that with just five users, about 80% of a product's usability issues can be highlighted (Nielsen 2000).

In addition, for the evaluation process, users were asked to perform the standard application usage scenarios shown in Table 8.

Interviews were then conducted, which were based on the following list of questions, but of course not limited to it:

1. Did you find the actions you had to take easy or difficult?
2. What did you think of the overall design of the app?
3. Do you think something extra should be added to the app?
4. What did you like the most?
5. Are there any other comments you would like to make?

In fact, as was the case with the Talking Subject method, users responded that they liked the soft colors of the app and that there was no information that was tiring to the eye. They also mentioned that they found all the steps they took simple and understandable and that the progress bar when choosing courses, reviewing and paying is very nice. Therefore, we conclude that the non-functional requirements of the app for intuitive use and tasteful design have been met. In addition, the findings of the assessment with both the Talking Subject method and the interviews are shown in aggregate in Table 9.

*Table 9. Findings of the evaluation with users using the method of interviews and the Talking Subject Protocol.*

n/a	Finding
1	The soft colors of the application are aesthetically beautiful.
2	Proper use has been made of white space.
3	There is no unnecessary information that tires the eye.
4	It would be desirable when selecting the dates in figure 28 to show which dates there are courses available.
5	The steps that need to be taken for the various actions are simple and understandable.
6	The progress bar is very nice when choosing courses, overviewing, and paying.

### 7.3.3 UEQ short questionnaire

An easy method to quantify how well designed the app is is the questionnaire method. In fact, we use this method because we aim to understand whether non-functional requirements for elegant and intuitive design have been met and to understand how users generally judge their interaction with the app.

For this purpose, the UEQ short questionnaire (Hinderks et al. e.g.) was used. In fact, this questionnaire was used for two reasons. First, it contains only eight questions, so it won't tire users, some of whom even participated in the interviews. Second, this questionnaire is designed to quantify how users rate the real-life part of the app as well as the aesthetic part of the app. Therefore, it is ideal for verifying the non-functional requirements we have set.

Specifically, the questionnaire consists of eight pairs of opposing words that describe the application (e.g. complicated-easy) and between them there is a linear climate with seven choices. The entire questionnaire is shown in Figure 69. We notice that the first four questions are about realistic amounts about the application, such as whether it is complex, clear, and supportive. The last four questions are about the emotions that the app creates in the user, such as whether it is indifferent or interesting.

disorienting	o o o o o o o	Supportive
Complicated	o o o o o o o	Easy
insufficient	o o o o o o o	sufficient
Confusing	o o o o o o o	clear
Boring	o o o o o o o	Exciting
Indifferent	o o o o o o o	Interesting
Conventional	o o o o o o o	Inventive
common	o o o o o o o	Innovative

Image 69. UEQ short questionnaire.

In fact, the creation and distribution of the questionnaire was done with the Google Forms tool (Google. e.g.). In addition, for the evaluation process, users were asked to carry out a typical scenario of using the app, i.e. to schedule a lesson for the next Christmas holidays at the Parnassos ski resort. Furthermore, the sample that responded to the questionnaire has a size of N=15.

To process the data, the scale of 7 options is transformed into scores from -3 to +3. However, it is reported that scores below -0.8 indicate poor design, from -0.8 to +0.8 neutral design and above +0.8 good design (Hinderks et al. e.g.). Of course, it is emphasized that scores at the ends of the scale, i.e. below -2 and above +2, are rarely observed due to the fact that users usually avoid scoring extreme choices on the linear scale.

Below are the results of the questionnaire assessment.

Table 10. UEQ short questionnaire results.

n/a	Average price	Dispersion	Scale
Question 1	1,8	0,4	Realistic
Question 2	1,8	1,2	Realistic
Question 3	1,7	1,0	Realistic
Question 4	1,7	1,0	Realistic
Question 5	1,3	0,5	Aesthetics
Question 6	1,3	1,0	Aesthetics
Question 7	1,2	1,1	Aesthetics
Question 8	1,0	1,2	Aesthetics

Table 11. Aggregate short UEQ results.

Scale	Average price
Realistic	1,722
Aesthetics	1,222
Totally	1,472

Initially, we notice that the average value in all questions is above +0.8 which indicates good design of the application. Of course, in Table 10 we see that the questions concerning the realistic aspects of the application have an extremely good score with an average value equal to or greater than +1.7. Also, questions about users' emotions, such as whether the app is exciting, score well, above +1.2, but slightly lower than the questions about the realistic aspects of the app. This is to be expected as the Easy Snow app mainly aims to improve the process for scheduling lessons in the ski resorts of Greece in a direct, easy and intuitive way and the main goal of the application is not to evoke emotions in users. In any case, however, the average value from all questions is 1,472, which is very good. In fact, on a scale with a maximum of +3, +1,472 may not seem such a good value, but as the authors of the questionnaire (Hinderks et al., etc.), it is actually a very good score as users usually avoid extreme values when scoring.

In addition, the variance in prices expresses the variation in scores in user responses. We notice that in general the questions have a low dispersion around 1, meaning that users generally agreed on their answers.

In conclusion, the average value in the questionnaire questions is very good and confirms to us that the non-functional requirements set regarding the intuitive and tasteful design of the application have been achieved. Also, users positively rate their overall interaction with the app.

## 7.4 Changes

As mentioned in the evaluation with the Talking Subject Protocol, finding 4 of Table 9 emerged, according to which it would be desirable for the application to show students which dates are available and which are not available when selecting course dates. Therefore, an additional function has been added, which takes into account the choices made by the user about the ski resort, the sport and the number of participants, and only shows the dates when there are available lessons. The initial and final implementation is shown in Figures 70 and 71 respectively.

In fact, in Figure 70 it can be seen that initially in the date selection the user can select any date. Figure 71 shows the date selection menu after addition, where users can now know from the beginning which dates there are courses available.

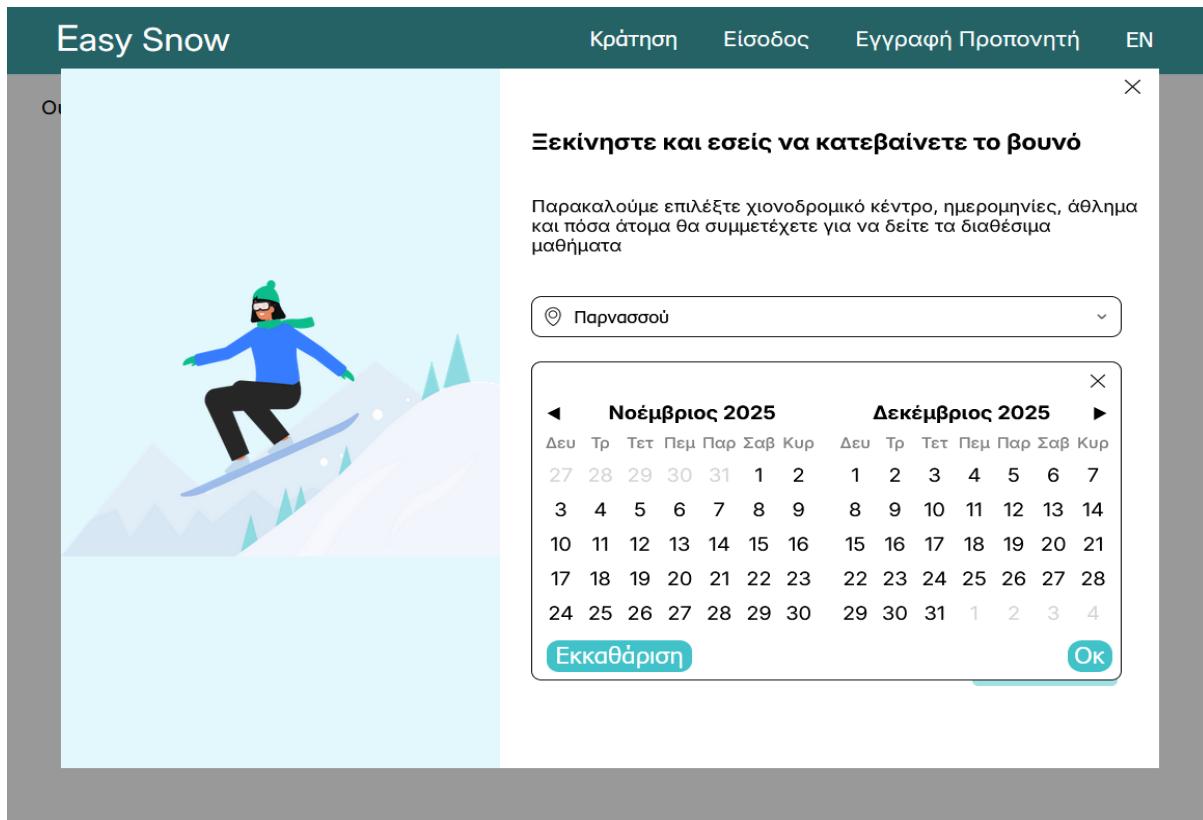


Image 70. Initial calendar implementation where the user can select all dates.

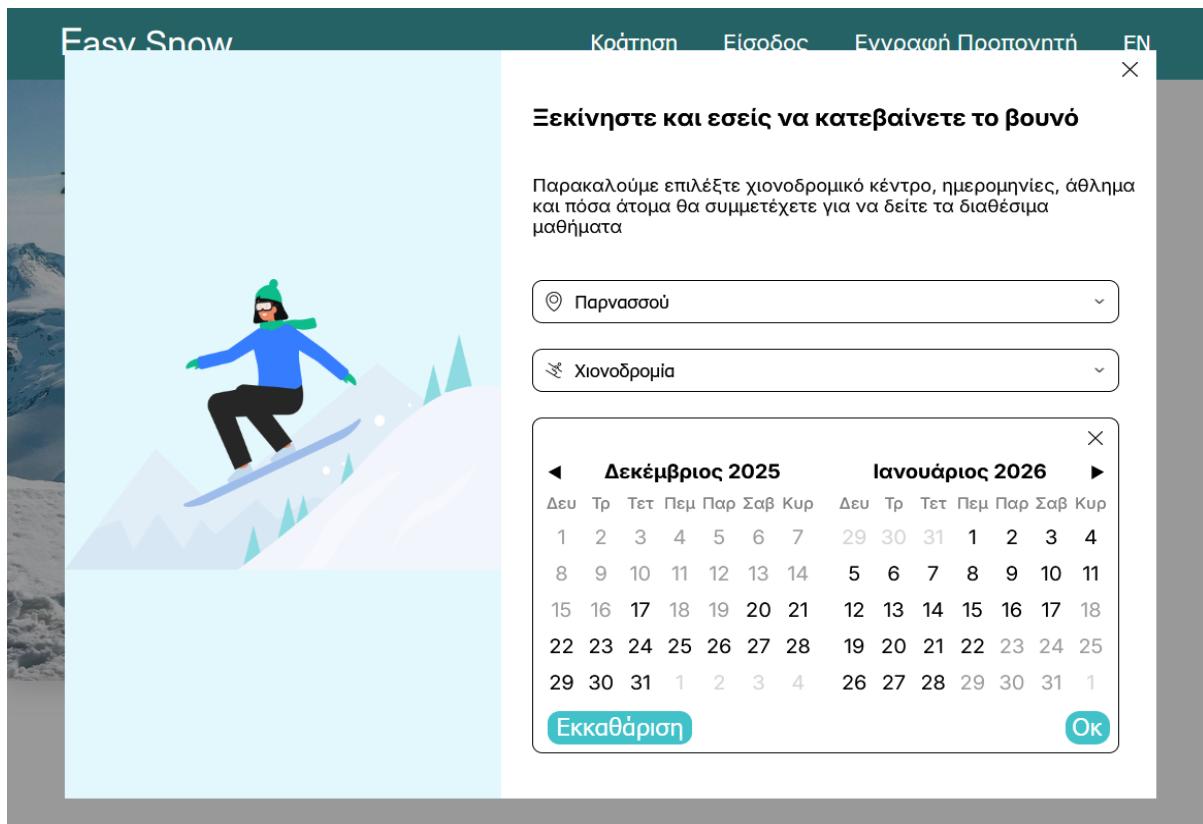


Image 71. Improved calendar implementation where the user is shown only the dates when there are available courses.

## 7.5 Conclusions

In this chapter, an overall evaluation of the application was carried out using evaluation methods by experts and users. Heuristic evaluation methods and Simplified Cognitive Touring were used as evaluation methods by experts, which did not reveal any usability problem. Also, the expert evaluation showed that the functional requirements of Table 3 have been met. In fact, the Talking Subject Protocol, semi-structured interviews were used as end-user assessment methods and users were asked to complete the simplified UEQ questionnaire. From the above process it was understood that the non-functional requirements of Table 4 for intuitive use of the application and elegant design have been achieved. Also, some users highlighted an issue with the app's usability that could be improved. In particular, when choosing dates for course scheduling, it would be good to show which dates are available and which are not. Finally, this plugin was implemented and integrated into the Easy Snow app, to ensure the optimal user experience.

# 8 Conclusions

In this thesis, the analysis, design, implementation and evaluation of an application that facilitates the scheduling of winter sports courses in ski resorts in Greece was carried out. In fact, the application aims to improve the current situation in Greece, where the scheduling of classes is mainly done by telephone. This goal was achieved as the Easy Snow app was created, which fully supports the needs of students and coaches.

## 8.1 Results

In section 1.1 the purpose of this diploma has been defined as

"... the design, development and evaluation of an application that enables winter sports coaches, regardless of whether they belong to a school or not, to offer lessons at any ski resort in Greece and at the same time allows skiers to choose the coaches that best meet their needs."

To sum up, we conclude that this goal has been fully achieved as the Easy Snow app was created, which provides all the necessary features to support skiers and coaches. Besides, in section 4.3 the functional and non-functional requirements of the application were recorded, and according to the design, implementation and evaluation process that followed, it emerged that they have been fully implemented. In particular, both the expert evaluation and the user evaluation showed that the Easy Snow application meets the requirements set, while offering a very good user experience.

## 8.2 Summary

In this thesis, all the individual steps for the creation of an application were implemented. Specifically, the current ways of scheduling winter sports courses in Greece and abroad were initially studied, where it was observed that there were basic shortcomings that the Easy Snow application had to solve. Then, anthropocentric planning, which is a fundamental element of this work, was analyzed. Then, the typical users were studied and analyzed, the technology with which users will interact with the app, the actions they will take and the context of use of the app. From the above analysis, the PACT framework was defined as well as the functional and non-functional requirements of the application.

Then prototypes were designed for the application as they are a cheap and fast way to study alternative design options. In fact, the colors of the application, the logo and the general appearance were specified. In addition, all icons and scribbles that will be used have been identified. Also, detailed prototypes were created that represent all the individual pages, which allows to ensure that consistency between the different pages is maintained and how the optimal design solution is selected each time. Moreover, with the completion of the prototype design process, a first phase of formative evaluation with end-users followed, using the Talking Subject

method. This process resulted in changes, which were incorporated into the final plans of the implementation.

Having completed the design process, the implementation of the application followed. For this purpose, React was used, which is one of the most well-known and modern libraries for creating modern graphical interfaces. Specifically, React was used to create a Single Page Application (SPA), in which all html code is built on the client side via Javascript. In addition, the React Router library was used to manage the history and quickly load the data from the server.

The application was then evaluated by both experts and end users. Specifically, the methods of Simplified Cognitive Reading and Heuristic evaluation were used for the evaluation by experts. Also, the end-user assessment was done through the Talking Subject Protocol, semi-structured interviews and the UEQ short questionnaire. In fact, the evaluation highlighted that the application is properly designed and fully meets the functional and non-functional requirements of the application. Furthermore, the comments from the end users were very positive and, correspondingly, the score that emerged from the questionnaire was very good. Of course, the evaluation revealed an element that could be improved in the app to ensure an even better user experience, which was implemented and integrated into the app.

### 8.3 Future steps

The Easy Snow app is a comprehensive effort to modernize the way winter sports are scheduled in ski resorts. Therefore, the next step in the development would be its use in real conditions by the thousands of students who every year hold lessons with coaches in the ski resorts of Greece. Of course, the app can be extended just as well to other European and non-European countries to create a vibrant, large and active community of skiers and coaches around the world. However, additional needs may arise from the daily use of the application, which are not included in the scope of this work. These needs could be implemented in the future to ensure the best possible user experience at all times.

## 9 Bibliography

- Avouris, N., Katsanos X., Tselios N., & Moustakas K. (2018). *Introduction to Human-Computer Interaction*. University of Patras Publications
- Abramov, d. (2021). *Before you memo()*. overreacted. <https://overreacted.io/before-you-memo/>
- altexsoft. (2023). *Functional and Nonfunctional Requirements: Specification and Types*. <https://www.altexsoft.com/blog/functional-and-non-functional-requirements-specification-and-types/>
- Angular. (ch.c.). <https://angular.dev>
- Anie, A. (2023). *Link Component in React Router*. Medium. <https://medium.com/@alexanie/link-component-in-react-router-6872291bf78e>
- Banthis, B. (2023). *PostgreSQL – concepts, benefits and use cases*. Tessell. <https://www.tessell.com/blogs/postgresql-concepts-benefits-and-use-cases>
- Benyon, D. (2019). *Designing user experience : a guide to HCI, UX and interaction design* (4th ed.). Pearson Education Limited.
- BrowserStack. (2025). *Common Screen Resolutions in 2025: Mobile, Desktop & Tablet*. <https://www.browserstack.com/guide/common-screen-resolutions>
- Chapman, C. (2021). *Color Theory for Designers, Part 1: The Meaning of Color*. Smashing Magazine. <https://www.smashingmagazine.com/2010/01/color-theory-for-designers-part-1-the-meaning-of-color/>
- Chart.js. (c.c.). <https://www.chartjs.org>
- Cite, Y. (2025). *React Router 7: Prefetching and Optimistic UI*. Medium. <http://levelup.gitconnected.com/react-router-7-prefetching-and-optimistic-ui-9a71ccc969fa?gi=957e9275c9e7>
- Cooper, A. (2004). The inmates are running the asylum: Why high-tech products drive US crazy and how to restore the sanity. Sams.
- Createley. (2024). *What Is an Entity Relationship Diagram (ERD)? | An Introduction to ER Diagram*. <https://createley.com/guides/er-diagrams-tutorial/>
- Deshpande, C. (2025). *The best guide to know what is react*. Simplilearn. <https://www.simplilearn.com/tutorials/reactjs-tutorial/what-is-reactjs>
- Developer Way. (2023). *Preventing React re-renders with composition*[Video]. YouTube. <https://www.youtube.com/watch?v=7sgBhmLjVws>

- Devkar, M. (2025). *Mastering React ScrollRestoration For Better Navigation*. DhiWise. <https://www.dhiwise.com/blog/design-converter/mastering-react-scrollrestoration-for-better-navigation>
- Djurdeh, H. (2024). *Server-side routing vs. client-side routing*. Telerik Blogs. <https://www.telerik.com/blogs/server-side-routing-vs-client-side-routing>
- Eagles, L. (2022). *How Javascript works: A deep dive into Vite*. Medium. <https://medium.com/sessionstack-blog/how-javascript-works-a-deep-dive-into-vite-965bdd8ff42>
- EmailJS. (ch.c.). *Overview*. <https://www.emailjs.com/docs/tutorial/overview/>
- ERD Maker. (2021). <https://hci.ece.upatras.gr/erdmaker/designer>
- Erolin, J. (2023). *React single page application*. BairesDev. <https://www.bairesdev.com/blog/react-spa-single-page-application/>
- Express. (ch.c.). <https://expressjs.com>
- Google. (ch.c.). *Google Forms*. <https://forms.google.com/>
- Hartson, H. R., & Hix, D. (1986). *Human-Computer Interface Development: Concepts and systems for its management*. Dept. of Computer Science, Virginia Polytechnic Institute and State University.
- Helmet. (ch.c.). <https://www.npmjs.com/package/helmet>
- Hinderks, A., Schrepp, M., & Thomaschewski, J. (x.x). *User Experience Questionnaire (UEQ)*. <https://www.ueq-online.org/>
- Hornsby, P. (2010). *Hierarchical Task Analysis*. UXmatters. <https://www.uxmatters.com/mt/archives/2010/02/hierarchical-task-analysis.php>
- i18next. (ch.c.). *Introduction*. <https://www.i18next.com>
- Ilot, M. (2023). *Password Hashing and Storage Basics*. Medium. <https://markilotto.medium.com/password-storage-basics-2aa9e1586f98>
- International Organization for Standardization. (2019). *Ergonomics of human-system interaction (ISO Standard No.9241-210:2019)*. <https://www.iso.org/obp/ui/#iso:std:iso:9241:-210:ed-2:v1:en>
- Inyang, I. (2024). *Bundle Up? Vite or Webpack?* Medium. <https://medium.com/@iboroinyang01/bundle-up-vite-or-webpack-c260915e0ff7>
- Jacob, S. (2023). Securing Your Node.js Application Security with Helmet. Medium. <https://medium.com/@selieshjksofficial/securing-your-node-js-application-security-with-helmet-296377385d07>

- Kent, D. (2019). *One simple trick to optimize re-renders*. Kentcdodds. <https://kentcdodds.com/blog/optimize-react-re-renders>
- Keshri, K. (2023). *The Crucial Role of Bundlers in React Development*. Medium. [https://medium.com/@krishnakeshri\\_30423/the-crucial-role-of-bundlers-in-react-development-bca6c67c1681](https://medium.com/@krishnakeshri_30423/the-crucial-role-of-bundlers-in-react-development-bca6c67c1681)
- Lewis, C. H. (1982). *Using the "thinking aloud" method in cognitive interface design* (Technical Report No. RC-9265). IBM.
- Lewis, C., Polson, P. G., Wharton, C., & Rieman, J. (1990). *Testing a walkthrough methodology for theory-based design of walk-up-and-use interfaces*. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems Empowering People - CHI '90, 235-242. <https://doi.org/10.1145/97243.97279>
- Lonc, J. (2024). *What is Prototyping in Software Development (and Why It's Important)*. SPARK Business Works. <https://sparkbusinessworks.com/blog/prototyping-in-software-development>
- MDN web docs. (ch.c.). *CSS: Cascading Style Sheets*. <https://developer.mozilla.org/en-US/docs/Web/CSS>
- Meta. (ch.c.). *React*. React.dev. <https://react.dev>
- Morville, P. (2016). *User experience honeycomb*. Intertwingled. <https://intertwingled.org/user-experience-honeycomb/>
- Nielsen, J. (1993). *Usability Engineering* (pp. 26–36). Academic Press.
- Nielsen, J. (2000). *Why You Only Need to Test with 5 Users*. Nielsen Norman Group. <https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/>
- Nielsen, J. (2017). *A 100-year view of user experience*. Nielsen Norman Group. <https://www.nngroup.com/articles/100-years-ux/>
- Nodejs. (ch.c.). *About Node.js*. <https://nodejs.org/en/about>
- Norman, D. (2018). *Principles of Human-Centered Design*[Video]. YouTube. <https://www.youtube.com/watch?v=rmM0kRf8Dbk>
- PostgreSQL. (ch.c.). *Unique Indexes*. <https://www.postgresql.org/docs/current/indexes-unique.html>
- Powney, A. (2024). *70% of IT, Technology, and Digital projects fail*. LinkedIn. <https://www.linkedin.com/pulse/70-technology-digital-projects-fail-andrew-powney-kmhae>
- React Router. (ch.c.). *React Router Home*. <https://reactrouter.com/home#react-router-home>

Remix Run. (ch.c.). *auth-router-provider* example. StackBlitz.  
<https://stackblitz.com/github/remix-run/react-router/tree/main/examples/auth-router-provider?file=README.md>

Rowley, D. E., & Rhoades, D. G. (1992). *The cognitive jogthrough*. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '92, 389-395.  
<https://doi.org/10.1145/142750.142869>

Stack Overflow. (2024). *Most popular technologies: Database*. 2024 Developer Survey.  
<https://survey.stackoverflow.co/2024/technology#most-popular-technologies-database>

SurveyJS. (ch.c.). *Client-Side vs Server-Side Form Input Validation*. <https://surveyjs.io/stay-updated/blog/client-server-data-validation>

Talmi, Y. (2023). What is a Helmet Content Security Policy, and Do You Need It? Cybeready.  
<https://cybeready.com/helmet-content-security-policy/>

Teixeira, F. (2016). *The right tool for the job: Picking the Best Prototyping Software for Your Project*. Smashing Magazine. <https://www.smashingmagazine.com/2016/06/picking-the-best-prototyping-software-for-your-project/>

Vue.js. (c.e.). <https://vuejs.org>

Web Dev Simplified. (2022). *Speed Up Your React Apps With Code Splitting[Video]*. YouTube.  
[https://www.youtube.com/watch?v=JU6sl\\_yyZqs](https://www.youtube.com/watch?v=JU6sl_yyZqs)

White, M. (2023). *How tough is bcrypt to crack? And can it keep passwords safe?* Specopssoft.  
<https://specopssoft.com/blog/hashing-algorithm-cracking-bcrypt-passwords/>

Zanini, A. (2023). Using Helmet in Node.js to secure your application. Logrocket.  
<https://blog.logrocket.com/using-helmet-node-js-secure-application/>