

Semester Project Documentation (DSA – CS – 221)

Semester Project Title: GIKI-map

Student Details:

	Student Name	Student Reg #	Student Degree
Student-1	Daniyal Qasim	2024147	BSFEE
Student-2	Abbas Tapal	2024005	BSFEE
Student-3	Ahsan Khalid	2024072	BSFEE

1.Main Features

Campus Location Mapping System

Shortest Path Calculation Between Locations

Route History Management

Sorted Route Analysis (Ascending & Descending)

Hierarchical Storage of Locations Using AVL Tree

2.Types of Users & Requirements

1. Student / Visitor (End User)

1. Will be able to select a starting location from the campus map.
2. Will be able to select a destination location.
3. Will be able to view the shortest distance between two locations.
4. Will be able to view the complete path between locations.
5. Will be able to view previous navigation history.

2. System (Navigation Engine)

Will calculate shortest path using Dijkstra's Algorithm.

1. Will store route history in a persistent file.
2. Will sort route history based on distance.
3. Will maintain ascending and descending route views.

3. Admin / System Manager

1. Will be able to manage campus locations.
2. Will be able to update distance connections between locations.
3. Will be able to monitor navigation history stored in files.

3. Requirements Breakdown

Feature 1: Campus Location Mapping

- 1.1 Store all campus locations in a vector.
- 1.2 Represent campus paths using a weighted adjacency list.
- 1.3 Maintain bi-directional connections between locations.

Feature 2: Shortest Path Calculation

- 2.1 Take source and destination as input from user.
- 2.2 Apply Dijkstra's Algorithm to calculate minimum distance
- 2.3 Display shortest path and total distance.

Feature 3: Route History Management

- 3.1 Store each shortest route in a text file.
- 3.2 Retrieve stored routes on program execution.
- 3.3 Maintain route data in structured format.

Feature 4: Sorted Route Analysis

- 4.1 Sort route history in ascending order of distance.
- 4.2 Store ascending order using Queue data structure.
- 4.3 Store descending order using Stack data structure.

Feature 5: AVL Tree-Based Location Storage

- 5.1 Insert all campus locations into AVL Tree.
- 5.2 Perform balancing using rotations.
- 5.3 Display tree using level-order traversal.

Sr #	Feature Name	DSA Concepts Used	Operations	Approx. Complexity	Variables / Objects	Functions	Lines of Code
1	Campus Mapping	Graph (Adjacency List), Vector	Insertion, Traversal	$O(V + E)$	15	2	~120
2	Shortest Path	Graph, Dijkstra Algorithm	Traversal, Comparison	$O(V^2)$	12	1	~90
3	Route History	File Handling, Vector	Read, Write, Traversal	$O(n)$	8	2	~60
4	Sorted History View	Stack, Queue, Sorting	Sorting, Insertion	$O(n^2)$	10	3	~75
5	Location Storage	AVL Tree	Insertion, Rotation	$O(\log n)$	14	5	~130

Project screenshots:

```
enter starting location(0-33): 13
enter ending location(0-33): 1
Shortest distance: 6.5
Path: FBS -> mess -> mosque -> hostel_2 -> hostel_9
```

```
Visual Tree Structure (showing each root left child and right child for each subnote of the level of tree)
Node hostel_10 =  left child admin_block, right child hostel_4

Node admin_block =  left child FBS, right child futsal
Node hostel_4 =  left child hostel_2, right child logic

Node FBS =  left child ACB, right child FMCE
Node futsal =  left child brabers, right child gym
Node hostel_2 =  left child hostel_12, right child hostel_3
Node logic =  left child hostel_9, right child mess

Node ACB =  left = null ,  right child ATM
Node FMCE =  left child FEE, right child FME
Node brabers =  left child auditorium, right child circket_ground
Node gym =  left child gate, right child hostel_1
Node hostel_12 =  left child hostel_11, right child hostel_13
Node hostel_3 =  left = null ,  right = null
Node hostel_9 =  left child hostel_7, right child library
Node mess =  left child medical_center, right child mosque
```

```
Node ATM = left = null , right = null
Node FEE = left = null , right = null
Node FME = left = null , right child Tuc
Node auditorium = left = null , right = null
Node circket_ground = left = null , right child faculty_club
Node gate = left = null , right = null
Node hostel_1 = left = null , right = null
Node hostel_11 = left = null , right = null
Node hostel_13 = left = null , right = null
Node hostel_7 = left child hostel_5, right child hostel_8
Node library = left child incubation_center, right = null
Node medical_center = left = null , right = null
Node mosque = left = null , right child sports_complex

Node Tuc = left = null , right = null
Node faculty_club = left = null , right = null
Node hostel_5 = left = null , right = null
Node hostel_8 = left = null , right = null
Node incubation_center = left = null , right = null
Node sports_complex = left = null , right = null
```

```

hostel_10 -> (hostel_9, 3) (hostel_1, 2) (gym, 2.5)
hostel_9 -> (hostel_10, 2.5) (futsal, 1) (circket_ground, 1) (hostel_2, 2)
hostel_1 -> (hostel_10, 2) (hostel_2, 1)
gym -> (hostel_10, 2.5) (sports_complex, 1) (futsal, 1.5)
sports_complex -> (gym, 1)
futsal -> (gym, 1.5) (hostel_9, 1) (circket_ground, 1)
circket_ground -> (futsal, 1) (hostel_9, 1) (mosque, 3) (admin_block, 4)
hostel_2 -> (hostel_9, 2) (hostel_1, 1) (mosque, 2)
mosque -> (hostel_2, 2) (circket_ground, 3) (mess, 1.5)
admin_block -> (circket_ground, 4) (admin_block, 1) (FEE, 1.5) (logic, 1.2)
gate -> (gate, 1)
mess -> (mosque, 1.5) (hostel_5, 2.5) (FBS, 1) (hostel_8, 3)
hostel_5 -> (mess, 2.5) (hostel_4, 1)
FBS -> (mess, 1) (hostel_8, 3.5) (ACB, 2) (FEE, 2)
hostel_4 -> (hostel_5, 1) (hostel_3, 2)
hostel_3 -> (hostel_4, 2) (hostel_8, 3) (hostel_11, 4.5)
hostel_8 -> (hostel_3, 3) (mess, 3) (hostel_11, 4) (hostel_12, 3) (FBS, 3.5) (ACB, 2)
hostel_11 -> (hostel_8, 4) (hostel_3, 4.5) (hostel_12, 2)
hostel_12 -> (hostel_11, 2) (hostel_8, 3) (ACB, 5)
ACB -> (hostel_8, 2) (hostel_12, 5) (FBS, 2) (library, 5.5) (auditorium, 3)
library -> (ACB, 5.5) (brabers, 2.8)
auditorium -> (ACB, 3) (FEE, 2.5) (FME, 1.7) (FMCE, 3.3)
FEE -> (auditorium, 2.5) (FBS, 2) (admin_block, 1.5) (FME, 2.2)
FME -> (FEE, 2.2) (auditorium, 1.7) (FMCE, 2.9)
logic -> (gate, 1.2) (ATM, 3.1)
ATM -> (logic, 3.1) (FMCE, 3.7) (Tuc, 3.1)
FMCE -> (ATM, 3.7) (FME, 2.9) (auditorium, 3.3) (brabers, 2.3)
Tuc -> (ATM, 3.1) (medical_center, 3.4)

brabers -> (library, 2.8) (FMCE, 2.3) (incubation_center, 0.6)
medical_center -> (Tuc, 3.4) (incubation_center, 1.4) (hostel_7, 0.4) (hostel_13, 1.7)
incubation_center -> (brabers, 0.6) (medical_center, 1.4)
hostel_7 -> (medical_center, 0.4)
hostel_13 -> (medical_center, 1.7) (faculty_club, 4)
faculty_club -> (hostel_13, 4)

```

```
history(unsorded):
gym --> medical_center : 17.3
sports_complex --> hostel_9 : 3.5
gym --> hostel_9 : 2.5
hostel_9 --> sports_complex : 3.5
sports_complex --> hostel_9 : 3.5
hostel_9 --> sports_complex : 3.5
sports_complex --> gym : 1
hostel_9 --> sports_complex : 3.5
gym --> circket_ground : 2.5
hostel_9 --> sports_complex : 3.5
hostel_9 --> futsal : 1
hostel_1 --> sports_complex : 5.5
sports_complex --> futsal : 2.5
sports_complex --> circket_ground : 3.5
sports_complex --> futsal : 2.5
sports_complex --> futsal : 2.5
sports_complex --> futsal : 2.5
gym --> sports_complex : 1
sports_complex --> futsal : 2.5
sports_complex --> futsal : 2.5
sports_complex --> futsal : 2.5
gym --> sports_complex : 1
hostel_9 --> gym : 2.5
hostel_7 --> hostel_9 : 16.3
circket_ground --> hostel_9 : 1
FBS --> hostel_9 : 6.5
FBS --> mess : 1
hostel_9 --> mess : 5.5
```

```
history in asending order(Queue):
sports_complex --> gym : 1
hostel_9 --> futsal : 1
gym --> sports_complex : 1
gym --> sports_complex : 1
circket_ground --> hostel_9 : 1
FBS --> mess : 1
sports_complex --> futsal : 2.5
sports_complex --> futsal : 2.5
gym --> circket_ground : 2.5
sports_complex --> futsal : 2.5
sports_complex --> futsal : 2.5
sports_complex --> futsal : 2.5
gym --> hostel_9 : 2.5
hostel_9 --> gym : 2.5
sports_complex --> futsal : 2.5
sports_complex --> futsal : 2.5
sports_complex --> hostel_9 : 3.5
hostel_9 --> sports_complex : 3.5
sports_complex --> circket_ground : 3.5
sports_complex --> hostel_9 : 3.5
hostel_9 --> sports_complex : 3.5
hostel_9 --> sports_complex : 3.5
hostel_9 --> sports_complex : 3.5
hostel_1 --> sports_complex : 5.5
hostel_9 --> mess : 5.5
FBS --> hostel_9 : 6.5
hostel_7 --> hostel_9 : 16.3
gym --> medical_center : 17.3
```

```
history in desending order(Stack):
gym -> medical_center : 17.3
hostel_7 -> hostel_9 : 16.3
FBS -> hostel_9 : 6.5
hostel_9 -> mess : 5.5
hostel_1 -> sports_complex : 5.5
hostel_9 -> sports_complex : 3.5
hostel_9 -> sports_complex : 3.5
hostel_9 -> sports_complex : 3.5
sports_complex -> hostel_9 : 3.5
sports_complex -> circket_ground : 3.5
hostel_9 -> sports_complex : 3.5
sports_complex -> hostel_9 : 3.5
sports_complex -> futsal : 2.5
sports_complex -> futsal : 2.5
hostel_9 -> gym : 2.5
gym -> hostel_9 : 2.5
sports_complex -> futsal : 2.5
sports_complex -> futsal : 2.5
sports_complex -> futsal : 2.5
gym -> circket_ground : 2.5
sports_complex -> futsal : 2.5
sports_complex -> futsal : 2.5
FBS -> mess : 1
circket_ground -> hostel_9 : 1
gym -> sports_complex : 1
gym -> sports_complex : 1
hostel_9 -> futsal : 1
sports_complex -> gym : 1
```

