1. **Introduction and Rationale**

The Hospital Bed Availability & Tracking System (HBATS) is intended to improve patient, visitor, and staff experiences while reducing hospital bed administration. It substitutes manual procedures that are prone to errors, gives visitors a safe way to find patients, and real-time visibility into bed occupancy across wards (ICU, General, Pediatric, Emergency). The system’s objectives are:

* Reduce patient wait times and better allocate resources.
* Improve patient and visitor privacy and convenience.
* Give hospital employees an effective digital tool to monitor the state of their beds.

1. **Content Evaluation**
2. **Navigation**

* **Staff Portal**: A safe space with dashboard, bed management, patient admission, and logout options.
* **Public Portal**: Visitor lookup on a single page without complex navigation.

1. **Layout & Wireframes**

* **Staff Dashboard:** Responsive grid showing color-coded beds (Green: Available, Red: Occupied).
* **Visitor Lookup:** Centered form with “patient Name” and “Passcode” fields.
* **Wireframes:** In Week- 3 Figma wireframes for all key pages: Admin Dashboard, Staff Bed View, Patient Passcode generator, Visitor Lookup will be covered.

1. **Content descriptions and sources**

* All text content and simple SVG icons will be created by the developer.
* Dummy data for wards, beds, patients, and staff will be generated via SQL scripts.

1. **Content-Types**

**Ward:** Hospital departments including ICU, General.

**Bed:** Individual bed within a ward with status (available/occupied).

**Patient**: Individual assigned to a bed.

**Staff:** Hospital employee with login credentials.

**Passcode:** Secure code linked to a patient for visitor access.

1. **Functionality (User Stories)**
2. **Must Have User Stories**
3. **As an** Admin, **I want to** add and manage hospital wards, **so that** I can define the structure of the resources of the hospital.
4. **As an** Admin, **I want to** add beds to a particular ward, **so that** capacity of each ward is accurately reflected.
5. **As a** staff member, **I want to** log into a secure portal, **so that** I can access administrative functions.
6. **As a** staff member, **I want to** view a dashboard of all beds color-coded by status (available/occupied), **so that**  I can quickly access hospital capacity.
7. **As a** staff member, **I want to** update a status of bed to “occupied” when a patient is admitted, **so that** the dashboard reflects real-time occupancy.
8. **As a** staff member, **I want to** input a name of patients when admitting them, **so that** the patient is linked to the bed.
9. **As a** staff member, **I want to** update a status of bed to “available” upon patient discharge, **so that** the bed can be reassigned.
10. **As a** staff member, **I want to** help a patient to generate a unique visitor passcode, **so that** it can be shared with family.
11. **As a** visitor, **I want to** access a public lookup page, **so that** I can find the room of patient.
12. **As a** visitor, **I want to** enter the name and passcode of a patient, **so that** the system displays the correct room number upon a match.
13. **As a** system, **I want to** store all data in a MySQL database, **so that** information is persistent and secure.

**Hourly Estimates for Must Have User Stores:**

**Task Estimated Time(hrs) -----------------------------------------------------------------------------------------------------**

Build forms and API endpoints for managing wards 12

and beds (CRUD operations).

Create login form, session setup, validation, and 12

secure routing.

Develop dashboard to fetch data and display color 10

coded beds.

Implement UI buttons and logic to update bed status 8

Build forms and logic for patient admission and 14

passcode generation.

Create public visitor lookup page with form and 10

Validation API.

Set up MySQL database, tables, and connection 6

configuration. -----------------------------------------------------------------------------------------------------

**Total 72**

1. **Should Have User Stories**

**12. As a** staff member, **I want to** filter the dashboard view by specific ward type, **so that** I can focus on a specific department.

**13. As a** staff member, **I want to** filter the dashboard by bed status, **so that**  I can see only available or occupied beds.

**14. As a** system, **I want** validate user inpute and show error messages, **so that** users can correct mistakes.

**15. As an** Admin, **I want to** view a dashboard widget showing total hospital capacity and current occupancy, **so that** I can make decisions.

1. **As a** system, **I want to** authenticate staff users, **so that** only authorized personnel can access the portal.

**Hourly Estimates for Should Have User Stores:**

**Task Estimated Time (hrs) ----------------------------------------------------------------------------------------------------**

Develop client- side and server-side form validation 6

logic.

Implement frontend filter UI and modify backend 8

API for wards/status.

Create new dashboard widget with capacity calculation 5

logic.

Implement password hashing and secure session 8

management. ----------------------------------------------------------------------------------------------------

**Total 27**

1. **Nice To Have User Stories**
2. **As a** patient, **I want to** invalidate my previously generated passcode, **so that** I can control access if my visitors change.
3. **As an** Admin, **I want to** export a report of current bed occupancy, **so that** I can use it for offline analysis.
4. **As a** staff member, **I want to** reset my password via email, **so that**  I can regain access if I forgot it.
5. **As a** system, **I want to** be mobile-responsive, **so that** it works well on all devices.

**Hourly Estimates for Should Have User Stores:**

**Task Estimated Time (hrs) ----------------------------------------------------------------------------------------------------**

Build API endpoint and UI button to invalidate a 4

passcode.

Develop data formatting, file generation, and 6

download endpoint for reports.

Implement email integration and secure token flow for 10

password reset.

Apply CSS media queries and test responsive layout 8

on all devices. ----------------------------------------------------------------------------------------------------

**Total 28**

1. **Technical Specifications**
2. **Technology Stack**

**Frontend:** HTML, CSS, JavaScript

**Backend:** PHP

**Database:** MySQL

1. **Hosting Information:** This application will be hosted on a platform that supports PHP and MySQL, for example, Render.
2. **Version control:** GitHub
3. **Design Prototyping:** Figma
4. **Database Management:** PhpMyAdmin
5. **Data Design**
6. **Diagrams:**
7. **Ward Management Flow**

**A diagram of a system

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**2. Visitor Lookup Process**

**A diagram of a patient room

AI-generated content may be incorrect.**

**A screenshot of a patient login

AI-generated content may be incorrect.**

1. **User Authentication & Data Access Flow**

**A diagram of a system validates credential

AI-generated content may be incorrect.**

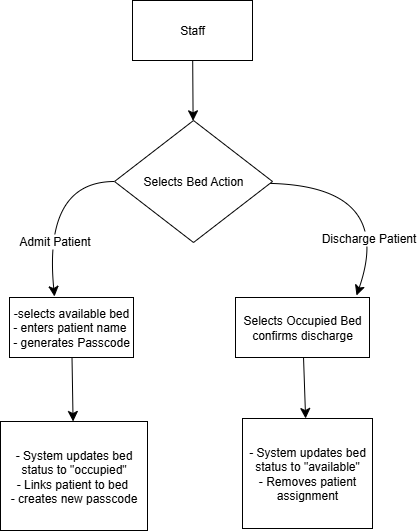
**A screenshot of a login page

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

1. **Bed Management Flow**

****

**A screenshot of a medical form

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

1. **Generate patient Passcode**

**A diagram of a passcode action

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

1. **Database Schema**
2. **wards**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| id | INT | Pimary Key, Auto\_Increment |
| name | VARCHAR (50) | Not Null |

1. **beds**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| id | INT | Pimary Key, Auto\_Increment |
| Bed Number | VARCHAR (100) | Not Null, Unique |
| status | ENUM | Not Null |
| ward\_id | INT | Foreign Key |

1. **patients**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| Id | INT | Pimary Key, Auto\_Increment |
| name | VARCHAR (100) | Not Null |
| assigned\_bed\_id | INT | Foreign Key, Unique |

1. **staff**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| Id | INT | Pimary Key, Auto\_Increment |
| username | VARCHAR (100) | Not Null, Unique |
| password\_hash | VARCHAR (100) | Not Null |
| role | ENUM | Not Null, Default “staff” |
| full\_name | VARCHAR (100) | Not Null |

1. **passcodes**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| Id | INT | Pimary Key, Auto\_increment |
| patient\_id | INT | Foreign Key |
| passcode | VARCHAR (100) | Not Null |
| expires\_at | DATETIME | Not Null |

**Relationships:**

* One ward can contain many beds.
* One bed can be assigned to one patient at a time.
* One patient can many passcodes.
* The staff table is standalone and contains user credentials for authentication. It does not have a direct foreign key relationship with the other tables in the core schema. Its relationship is through the application’ logic.

1. **Developer Specifications**
2. **Folder Structure**

Project-root/

admin/

staff/

public/

includes/

sql/

assets/

login.php/

logout.php/

index.php/

gitignore/

1. **File-naming Conventions**
2. **Kebab-case:** This format will be used for giving names to files, for example, staff-dashbpard.php, main-styles.php.
3. **Pascal case:** This format must be used for files, and the syntax is first letter of every word is capitalized, with no spaces or hyphens, for example, Ward.js/ Bed.**js.**
4. **camelCase:**  This format will also be used for giving names to files, with the first word being lowercase, and the first letter of every subsequent word is uppercase, with no spaces or hyphens, for example, dashboard.js.
5. **Responsiveness Requirements**

The application must be fully responsive and functional on all devices, including desktop, tablet, and mobile, screen sizes using CSS flexbox, grid and media queries.

1. **Timeline**

**Week Title Description Due Date Status**

**---------------------------------------------------------------------------------------------**

1. Complete capstone Finalize the idea 06-Sep-25 Completed

proposal document and get sign off

from instructor.

2. Complete requirements Completed detailed 12-Sep-25 Completed

document/timeline and project timeline.

technical barriers Finalized requirements

assessment. document. Indentified

& document potential

technical barriers &

solutions.

3. UX/UI Design & Create primary tables. 19-Sep-25 Completed

Database Planning Create figma wireframes

for all key pages.

Get feedback on figma

designs from professor.

4. Project setup Initialize PHP project 26-sep-25 Not started

structure. Create SQL

scripts.

5. Core Backend Build PHP forms & session 03-Oct-25 Not started

Logic & authentication. Figma User

Authentication testing.

User Testing

6. Frontend Code the frontend pages 10-0ct-25 Not started

Development, designs. Connect frontend

Integration& Bug forms to the Backend.

Fixing Test all features &

Fix Bugs..

7. Presentation. Present the final product 17-oct-25 Not Started

8. Final Deployment Ensure Live site is stable 26-oct-25 Not started

& Mentor meeting

**8. Conclusion**

The Hospital Bed Availability & Tracking System MVP will be developed using the detailed plan found in this document. The emphasis is on core functionality that shows proficiency in database design, full-stack JavaScript development, and user-centric problem-solving. This plan will direct development in an efficient and targeted manner.