# 

# 

# Hospital Database Management System

# 

CPS 714 - Software Project Management

Dr. Omar Falou

December 2, 2022

Section 1 Group 7

| **Name** | **Student Number** | **Email** |
| --- | --- | --- |
| Uzair Ahmed | 500905189 | u1ahmed@ryerson.ca |
| Anmol Sharma | 500808863 | anmol.sharma@ryerson.ca |
| A S M Rubayet Ahmed | 500962603 | asmrubayet.ahmed@ryerson.ca |
| Sean Samu | 500882775 | ssamu@ryerson.ca |
| Bill Wang | 501012196 | bill.wang@ryerson.ca |
| Hasyen Patel | 500951403 | hasyen.patel@ryerson.ca |
| Karthik Gurram | 500880724 | kgurram@ryerson.ca |
| Akash Patel | 500896220 | akash3.patel@ryerson.ca |
| Ankit Misra | 500898379 | ankit.misra@ryerson.ca |

# Table of Contents

[**Hospital Database Management System**](#_tyl7jsrs9ajg) **1**

[**Table of Contents**](#_mpj3yunokbs1) **2**

[**Part 1 - Introduction**](#_dnoc8ln9bgdr) **4**

[1.0 - Important Links](#_p2u75ylkuxro) 4

[1.1 - Purpose](#_x84x7g4amds1) 4

[1.2 - Scope](#_7mta36m86xq2) 4

[1.3 - Quality](#_uw9zo11ffa13) 5

[1.4 - Scheduling](#_aao7e1snvxav) 5

[1.5 - Communications](#_7xtcb1xvuwqh) 6

[Definitions](#_ojnbscjon37s) 7

[1.5 - Risks](#_hi3muek8hce) 7

[Methodology](#_j6l76bvmt77) 7

[Roles and Responsibilities](#_b8mhsa3knkz2) 7

[Budget and Schedule](#_907r54y83zc9) 7

[Risk Probability](#_ow86o8mogc0j) 7

[Risk Documentation](#_wmmlzetuvyog) 8

[1.6 - Cost Objectives](#_htrpj31d88ax) 8

[Cost Estimate](#_nujlty50dlim) 8

[1.7 - References](#_iaymjnappdi8) 9

[**Part 2 - Documentation**](#_wgh4kr574hrx) **11**

[2.0 - Project Charter](#_z09hux824h6y) 11

[2.1 - Stakeholders Register](#_daoi4zb09dw) 14

[2.2 - Meeting Minutes - Kickoff Meeting](#_qfigyhl275p0) 15

[2.3 - Project Management Tool (JIRA)](#_5a5hionvehxz) 16

[2.4 - Product Backlog](#_pn5g8k9alc97) 17

[Sprint 1](#_jrxwjm48abvf) 17

[Sprint 2](#_ufc0m9htwds6) 17

[Sprint 3](#_i2r1dfjasj62) 17

[Sprint 4](#_9or21y2dzh40) 17

[Sprint 5](#_v3jcguiuc3yi) 17

[2.5 - Sprint Backlog](#_3e9i1saj8d5j) 18

[Sprint 1](#_ichfl3gpkqjw) 18

[Sprint 2](#_vr9182heb8p5) 18

[Sprint 3](#_y96sxxk879kk) 19

[Sprint 4](#_8tpc1ep81p9v) 19

[Sprint 5](#_yg4848wk31qn) 20

[2.6 - Sprint Burndown Charts](#_b3v3hzg0hucj) 21

[Sprint 1](#_hd81oh84r008) 21

[Sprint 2](#_8ve0wjch002j) 21

[Sprint 3](#_7kghmj2m1h8c) 21

[Sprint 4](#_35jy3duyk15q) 22

[Sprint 5](#_y7hnbo7b5lp5) 22

[2.7 - Velocity report](#_vl8kpe9hikds) 23

[2.8 - Gantt Chart](#_tr86as61fzx7) 24

[2.9 - Meeting Minutes - Sprint Review](#_52hp2a393q6t) 25

[Sprint 4 Review](#_qs5m13fzvkcy) 25

[Sprint 5 Review](#_fdblseqc5n07) 26

[2.10 - Meeting Minutes - Sprint Retrospective](#_y2df00cebx5i) 27

[Sprint 4 Retrospective](#_2r5h28gl7lc8) 27

[Sprint 5 Retrospective](#_zhoxhxvhfoib) 28

[2.11 Source Code of Final Project](#_38n9wgtltrk6) 29

[Live Implementation](#_7c41lzzb3okf) 29

[Backend](#_zp38c6y1q1o) 29

[Frontend](#_ymg80cdsnde) 29

[**Part 3 - Technical Description and Documentation**](#_jhc6fut2ndoc) **30**

[3.0 - Product Perspective](#_641lk3t6237t) 30

[3.1 - Product Functions](#_d1rchzrnlo77) 31

# 

# Part 1 - Introduction

*This section gives a scope description and overview of everything included in this Final report. Also, the purpose for this document is described and a list of abbreviations and definitions is provided.*

## 1.0 - Important Links

Scrum Meeting Link: [Zoom](https://ryerson.zoom.us/j/5786345384?pwd=T1pUQjA2cHZzbVFzWEZDRWE4cXUxUT09)

JIRA Management: [JIRA - CPS714 Group 7](https://cps714-group7.atlassian.net/)

Bitbucket Frontend: [Bitbucket - Frontend](https://bitbucket.org/u1ahmed/frontend/src/master/)

Bitbucket Backend: [Bitbucket - Backend](https://bitbucket.org/u1ahmed/backend/src/master/)

Live Site URL: <https://cps-714-g7.netlify.app/>

Live Server URL: <https://cps-714-hospital-management.herokuapp.com/>

## 1.1 - Purpose

The purpose of this project is to create a Hospital Database Management System (DBMS) to allow for the digitalization of day-to-day processes performed in a hospital setting. Key functionality should include the ability to record patient and employee information, as well as appointment scheduling functionality. The Hospital DBMS should allow for improved efficiency for employees performing day-to-day tasks such as creating and editing patient and employee records, as well as creating and managing medical appointments. Functionality should ideally be limited based on user groups (physicians, receptionists, and patients): for example, a patient should not be able to view employee information. In accordance with their roles in the system, users should be able to access corresponding features.

## 1.2 - Scope

The “Hospital Management System” is a DBMS which helps hospital members (including patients) view information based on the user's current role at the hospital; all system information should be maintained in a centralized database. During login, users should be able to either create an account or log in using existing credentials, allowing for accurate access control to system features. Hospital employees such as doctors, physicians, nurses and administrators should be able to create, read, update, and delete patient information, appointment information, hospital equipment information, and hospital medication information using the web portal. In our design, we group all of the hospital employees that have access to this system into one user type to streamline the process. Hospital employees also have the ability to maintain employee information: for instance, they can add or remove employees and manage their personal information. Patients can view their personal information and any upcoming appointments. If a patient wishes to update their information, they would be required to contact a hospital employee to change it for them.

## 1.3 - Quality

“The quality of the product could be determined by the quality objectives, the criteria used to evaluate the project and product quality, and the verification and validation information.”

Product quality will be evaluated using several key quality objectives. At delivery, these main quality objectives must be attained: first, the number of defects in the product should be as low as possible, and it should not crash on execution. The second quality objective is that by developing the DBMS, system uptime should be improved to twenty-four hours per day; as a result, the hospital's operational efficiency should improve by eighty percent. A target of 100% on-time delivery rate is set for the development of this product. Thirdly, product requirements must be met, and the product should be functional and durable. The web portal should contain the functions and design which were requested from the specifications. The system should also be secure against attackers and data tampering. Lastly, the product should achieve a customer satisfaction rating of eighty percent.

Project quality is based on meeting the anticipated scope, timing, and cost objectives. Firstly, client feedback after delivery should be positive. Secondly, the project's primary goal of providing a system that hospital members can easily administer and use on a daily basis should be accomplished. The crucial factor for achieving verification is that documentation has been generated, maintained, and reviewed throughout all sprints. Project management tools should be correctly utilized in order to evaluate the needs and challenges of the project team: tools should be used to ensure that the project is being developed in accordance with the requirements, and whether the newly added functionality is appropriate or not. In order to obtain validation, the product should be tested thoroughly via test cases and mock data. Lastly, to assure validation, it should be ensured that the product is in line with the customer’s expectations.

## 1.4 - Scheduling

To define the team’s scheduling goals and track progress, implementation milestones are represented on a Gantt chart, including the planned and actual delivery dates. The first milestone was set on September 16, 2022, marking the project kickoff meeting where team members signed off on the project. The rest of the milestones or delivery dates were set at the end of each sprint: the second milestone date was October 7, 2022, marking the end of Sprint 1. October 21, 2022 marks the third milestone, as well as the conclusion of Sprint 2. The end of Sprint 3 — November 4, 2022 — marks the fourth milestone and another delivery date. The fifth milestone was scheduled for November 18, 2022, on completion of Sprint 4. The sixth and final milestone date was December 2, 2022, marking the conclusion of Sprint 5.

The Agile earned value management (EVM) technique is conveyed to illustrate the difference between measured and actual team performance. The main reason for the discrepancy between measured and actual performance was due to the lack of familiarity with newly introduced project tools such as Jira, Bitbucket, and Flask. Another factor that prevented the team from performing as expected was the time constraint we encountered in Sprint 3. The goal of the sprint was to connect frontend and backend processes, which was estimated to require a total of eighty story points worth of work. The volume of backlog items, combined with the hectic schedule of various team members, led to the outcome where the team was not able to complete all the tasks scheduled for the sprint. A similar discrepancy occurred during Sprint 4, where the activity durations for several tasks were estimated incorrectly, requiring significantly more work than anticipated. This led to the spillover of tasks to the following sprint.

## 1.5 - Communications

The management team will be monitoring the team to ensure that the project fulfills the sprint targets. The management team may accomplish this, for example, by adding tasks to the sprint dashboard and shifting them between the to-do, in-progress, and done subgroups. By November 22, 2022, the team should have finished coding the app's services, allowing it to be fully functional and ready for deployment before the sprint's completion. The provided Gantt chart will allow the team to see future plans for the upcoming release.

To ensure that the team is moving closer to the completed product, the stakeholders demand that the team have daily Scrum meetings throughout the sprints. The daily Scrum meetings should address what the team members did yesterday, what they plan on doing today, and if they are facing any impairments/blockers. The daily Scrum meetings should be recorded by the Scrum Master so that the team may review them as needed. If there is a limitation or a job that has to be completed by another member before one member can continue, the team will use the chosen project management application, JIRA, to write comments under the task to notify the other members.

At the closing of each sprint, meetings for retrospective and review should be held in order to draw lessons from the past and make plans for success in the coming sprint. The team's ability to improve for the next sprint or release, any concerns or problems they encountered during the current sprint, and tasks that still need to be completed should all be covered in the retrospective sessions. To maintain a moving workflow, the team should also commit to the daily scrum and retrospective processes.

### Definitions

| **Term** | **Definitions** |
| --- | --- |
| User | Someone who interacts with the database management system |
| Admin/Administrator | System administrator who is given specific permission for managing and controlling the system |
| DBMS | Database Management System |
| Web Portal | The web application interface which hospital members, including patients, will interact with |
| Stakeholder | Any person who has interaction with the system |

## 1.5 - Risks

### Methodology

Risk assessments are done to analyze and highlight risks, with the intent to identify and control said risks. Once high-risk areas are identified, risk control processes will be implemented.

### Roles and Responsibilities

Uzair Ahmed, Anmol Sharma and A S M Rubayet Ahmed are responsible for analyzing risks and identifying high risk areas.

### Budget and Schedule

Estimated cost of time is generally half of a sprint’s duration (one week).

### Risk Probability

Risk analysis is done to identify any potential risks. Qualitative risk analysis is performed using probability/impact matrix and top ten/five risk item tracking. The matrix is drawn by assigning high, medium and low scores to the probability and impact of certain risks, and aligning them on either sides of the axis. A risk can be considered high priority (coloured in red) if any of the impact or probability has a “high” score, implying that it will be given the most importance. Anything with a “low” score can be considered low priority and be coloured in green. All other risks which fall in between “high” and “low” scores will be coloured in yellow.

Quantitative risk analysis is done after qualitative risk analysis, where the risks identified in the previous analysis are each given a numerical value, and then documented. A higher rating denotes more importance.

### Risk Documentation

A risk log is maintained by the three team members, and during and at the end of each sprint the log is reviewed as an agenda item. The risks are tracked, monitored and reported throughout the project.

## 1.6 - Cost Objectives

### Cost Estimate

|  | Units/Hrs | Cost/Unit/Hr | Subtotals | Totals | % of Total |
| --- | --- | --- | --- | --- | --- |
| **1. Project Management** |  |  |  | $**80,000** | **12%** |
| Project Manager | 320 | $100 | $32,000 |  |  |
| Project Team Members | 640 | $75 | $48,000 |  |  |
| **2. Hardware** |  |  |  | $**18,646** | **3%** |
| 2.1 Servers | 5 | $3,729 | $18,646 |  |  |
| **3. Software** |  |  |  | $**418,000** | **62%** |
| 3.1 Licensed Software\*\* | 2 | $1,000 | $2,000 |  |  |
| 3.2 Software Development\* |  |  | $416,000 |  |  |
| **4. Testing (10% of total hardware and software costs)** |  |  | $43,664 | $**43,664** | **6%** |
| **5. Reserves (20% of total estimate)** |  |  | $112,062 | $**112,062** | **17%** |
| **Total project cost estimate** |  |  |  | $**672,372** |  |

*\* Formula:* (Subtotal of Project Team Members \* 8 Team Members) + (Subtotal of Project Manager)

*\*\* Google Firebase and Google Cloud hosting*

### 

| **1. Labor Estimate** | Units/Hrs | Cost/Unit/Hr | Subtotals | Calculations |
| --- | --- | --- | --- | --- |
| Project Team Members | 640 | $75 | $48,000 | 640\*75 |
| **Total labor estimate** |  |  | **$48,000** |  |
| **2. Function point estimate** | Quantity | Conversion Factor | Function Points | Calculations |
| External inputs | 40 | 4 | 160 | 40\*4 |
| External interface files | 3 | 7 | 21 | 3\*7 |
| External outputs | 3 | 5 | 15 | 3\*5 |
| External queries | 1 | 4 | 4 | 1\*4 |
| Logical internal tables | 4 | 10 | 40 | 4\*10 |
| **Total function points** |  |  | 240 | Sum above |
| Python |  |  | 46 | Assumed value |
| Source lines of code (SLOC) estimate |  |  | 11,040 | 240\*46 |
| Productivity x (KSLOC)^Penalty (in months) |  |  | 29.28 | 3.13\*8.05^1.072 |
| Total labor hours (27 hours/func. pt.) |  |  | 6,480 | 27\*240 |
| Cost/Labor hour ($80/hr) |  |  | $80 | Assumed value |
| **Total function point estimate** |  |  | **$518,400** | 6,480\*80 |

## 1.7 - References

Kathy Schwalbe (2019). Information Technology Project Management, 9th Edition. Cengage.

Robert K. Wysocki (2017). Effective Project Management: Traditional, Agile, Extreme, 7th edition. Wiley.

Adolfo Villafiorita (2013). Introduction to Software Project Management. Boca Raton, FL: CRC Press/Auerbach Publications.

Erik W. Larson and Clifford F. Gray (2017). Project Management: The Managerial Process. New York, NY: McGraw-Hill Higher Education.

David J. Anderson (2004). Agile Management for Software Engineering. New Saddle River, NJ: Pearson Education Inc

# Part 2 - Documentation

*This section includes all necessary documentation crucial to tracking project progress and product delivery.*

## 2.0 - Project Charter

| **Project Title** | | Group 7 Hospital Management System | |
| --- | --- | --- | --- |
| **Project Start Date** | 09/16/2022 | **Projected End Date** | 12/02/2022 |

| **Budget Information** | | 11 Weeks | |
| --- | --- | --- | --- |
| **Project Manager** | | Uzair Ahmed  (365) 777-4034  u1ahmed@ryerson.ca | |
| **Project Objectives** | | To develop a hospital database and create an application for both staff and patients to utilize, taking approximately eleven weeks to complete. The objective is to make it simple for users to incorporate the in-app capabilities that can be useful to them on a daily basis. Users may vary from physicians, patients, receptionists, or nurses, and they will receive features that are convenient to their job at the hospital.  Release Date: December 1, 2022  Maximum Duration: 11 weeks | |
| **Success Criteria** | | Our objective is to provide software that improves productivity and quality in the medical workplace. The program must pass rigorous testing and be able to carry out all of its planned capabilities in order to achieve this aim. The sprint review and retrospective sessions must be used to make long-term plans for success and on-time delivery. In order to achieve this aim, any pillars or concerns will be addressed during the daily scrum stand-up meetings. The Gantt chart and dashboards on Jira should be used as project management tools to keep track of progress and highlight any problems. The team will keep putting its whole effort into making superior, robust software. | |
| **Approach** | | The planned approach involves integrating the project management and system integration components to meet stakeholders' needs and expectations. To avoid suffering significant losses due to time constraints, the team will complete the most critical tasks first. Furthermore, the team will hold weekly project team meetings to review growth and put commitments from weekly planning sessions into action. The project plan will document and state the duties of team members. Lastly, every Scrum meeting will begin with addressing unsolved concerns regarding the assigned tasks.  Assumptions: Without proof or demonstration, assumptions are elements used for planning that are taken to be real or certain:   * No new hardware is required * Should function across all platforms and browsers   Constraints: The following conditions should be satisfied to prevent project limitations and thereby by meeting these conditions one should be able to monitor the project's time, cost, and scope.   * By 09/22/2022, system should be ready for development, allowing for complete design * Gantt chart should be completed by Sprint 4 * List of risks should be provided by Sprint 5 * The final product is expected to be completed and approved, within three months and two weeks of start date | |
| **Project Risks** | | Team does not have significant experience starting a project from scratch; it is therefore plausible that the system won't live up to the customer’s expectations, failing to achieve our goal. | |

| **Roles and Responsibilities** | | | |
| --- | --- | --- | --- |
| **Name and Initials** | **Role** | **Position** | **Contact** |
| Uzair Ahmed (UA) | Project Manager, Product Owner | Project Manager, Product Owner,  Frontend Developer,  Backend Developer | u1ahmed@ryerson.ca |
| Anmol Sharma (AS) | Scrum Master | Scrum Master | anmol.sharma@ryerson.ca |
| Akash Patel (AP) | Team Member | Backend Developer | akash3.patel@ryerson.ca |
| A.S.M. Rubayet Ahmed (RA) | Team Member | QA Tester | asmrubayet.ahmed@ryerson.ca |
| Sean Samu (SS) | Team Member | Backend Developer | ssamu@ryerson.ca |
| Hasyen Patel (HP) | Team Member | Frontend Developer | hasyen.patel@ryerson.ca |
| Karthik Gurram (KG) | Team Member | Backend Developer, Frontend Developer | kgurram@ryerson.ca |
| Ankit Misra (AM) | Team Member | Frontend Developer | ankit.misra@ryerson.ca |
| Bill Wang (BW) | Team Member | Backend Developer | bill.wang@ryerson.ca |

## 

## 2.1 - Stakeholders Register

| **Roles and Responsibilities** | | | | |
| --- | --- | --- | --- | --- |
| **Name** | **Role** | **Position** | **Contact** | **Internal/External** |
| Uzair Ahmed | Project Manager, Product Owner | Project Manager, Product Owner,  Frontend Developer,  Backend Developer | u1ahmed@ryerson.ca | Internal |
| Anmol Sharma | Scrum Master | Scrum Master | anmol.sharma@ryerson.ca | Internal |
| Akash Patel | Team Member | Backend Developer | akash3.patel@ryerson.ca | Internal |
| A.S.M. Rubayet Ahmed | Team Member | QA Tester | asmrubayet.ahmed@ryerson.ca | Internal |
| Sean Samu | Team Member | Backend Developer | ssamu@ryerson.ca | Internal |
| Hasyen Patel | Team Member | Frontend Developer | hasyen.patel@ryerson.ca | Internal |
| Karthik Gurram | Team Member | Backend Developer, Frontend Developer | kgurram@ryerson.ca | Internal |
| Ankit Misra | Team Member | Frontend Developer | ankit.misra@ryerson.ca | Internal |
| Bill Wang | Team Member | Backend Developer | bill.wang@ryerson.ca | Internal |
| Zahra Mohtajollah | Supervisor | - | zmohtajollah@ryerson.ca | External |
| Dr. Omar Falou | Supervisor | - | ofalou@ryerson.ca | External |
| Hospital Receptionist | User | - | - | External |
| Hospital Doctor | User | - | - | External |

## 2.2 - Meeting Minutes - Kickoff Meeting

**Kick Off Meeting**

**Date:** 08/16/2022

**Time:** 12:15 PM - 1:00 PM (45 Mins)

**Project Name:** Hospital Management System

**Meeting Objective:** Get the project off to an effective start by introducing key stakeholders, reviewing project goals, and discussing future plans

**Agenda:**

* Attendee Introductions
* Review of Project Background
* Review of Project Related Documents
* Discussion of Project Organizational Structure
* Discussion of Project Scope, Time, and Cost Goals
* Discussion of other important topics

**Daily Scrum Schedule:**

Monday - Thursday: 10:30 PM - 10:45 PM, Remote ([Zoom](https://ryerson.zoom.us/j/5786345384?pwd=T1pUQjA2cHZzbVFzWEZDRWE4cXUxUT09))

Friday: 12:15 - 12:30, In Person (ENG203)

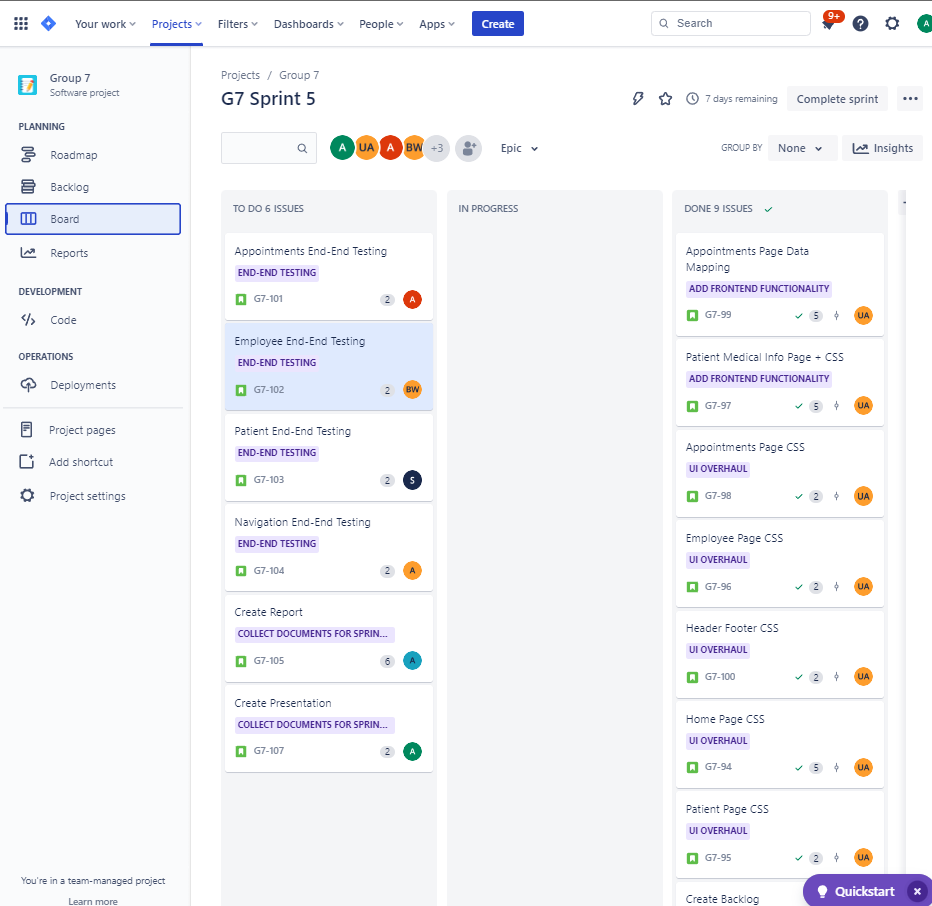
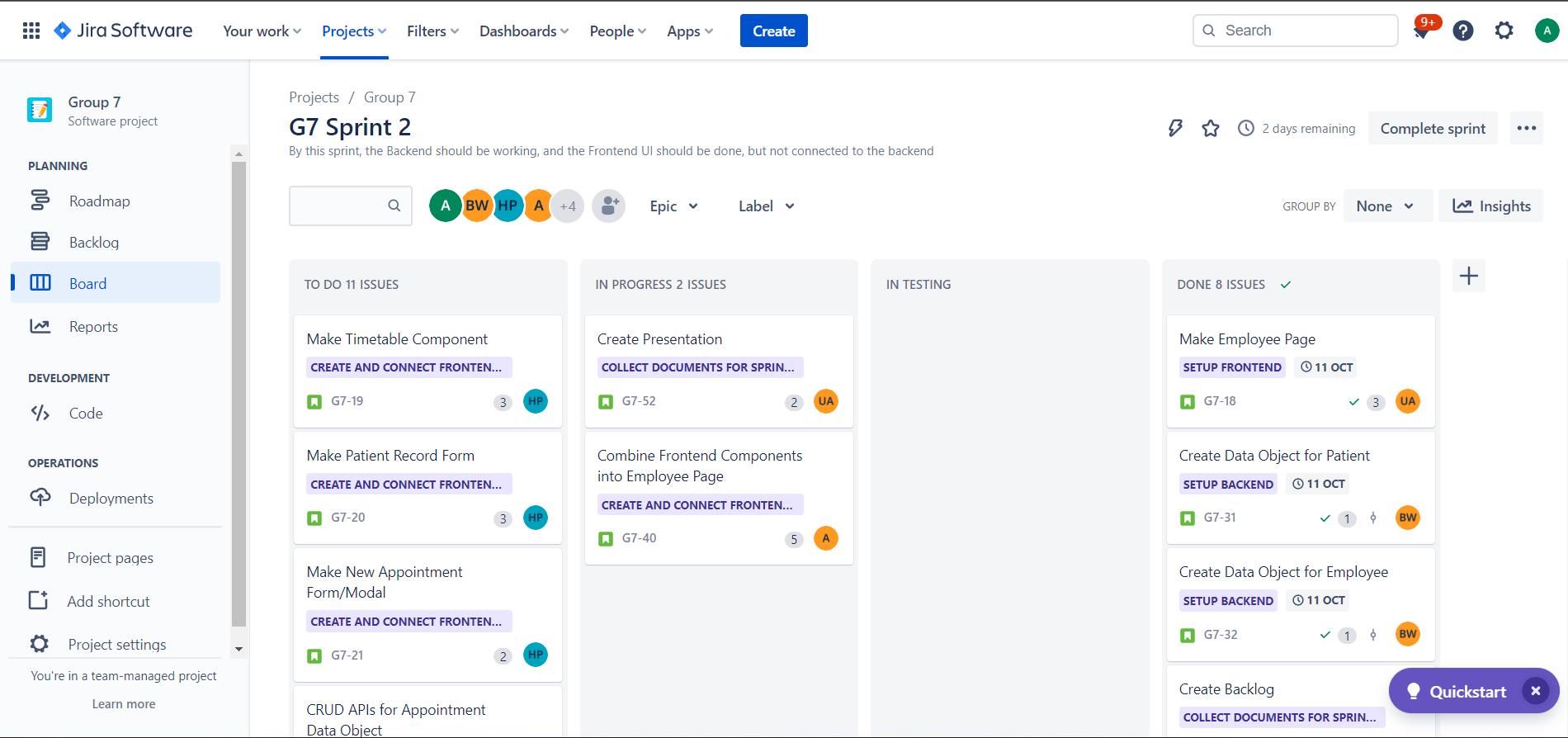
**Date and Time of Next Meeting:** 09/16/2022, 10:30 PM - 10:45 PM

**Project Management Tool:** JIRA (<https://cps714-group7.atlassian.net/>)

**Preliminary Development Action Items**

| **Action Item** | **Assigned To** | **Due Date** |
| --- | --- | --- |
| Assist in Dev Setup | Uzair Ahmed | 10/07/2022 |
| Collect Documentation | Anmol Sharma | 10/07/2022 |
| Setup NoSQL/Firebase | Akash Patel | 10/07/2022 |
| Setup NoSQL/Firebase | Sean Samu | 10/07/2022 |
| Setup Python/Flask | Karthik Gurram | 10/07/2022 |
| Setup Python/Flask | Bill Wang | 10/07/2022 |
| Setup Node/ReactJS | Ankit Misra | 10/07/2022 |
| Setup Node/ReactJS | Hasyen Patel | 10/07/2022 |
| Assist in Scrum | Rubayet Ahmed | 10/07/2022 |

## 2.3 - Project Management Tool (JIRA)



## 2.4 - Product Backlog

### Sprint 1

* Setup Frontend
* Setup Backend
* Collect Sprint 1 Documents

### Sprint 2

* Create and Connect Frontend Pages
* Create and Connect Backend Pages
* Collect Documents for Sprint 2

### Sprint 3

* Create and Connect Frontend Pages
* Create and Connect Backend Pages
* Improve Frontend Pages
* Add Frontend Functionality
* Collect Documents for Sprint 3
* QA Testing

### Sprint 4

* Add Frontend Functionality
* API Testing
* Deployment
* Collect Documents for Sprint 4

### Sprint 5

* UI Overhaul
* End-End Testing
* Collect Documents for Sprint 5

## 

## 2.5 - Sprint Backlog

### Sprint 1

* Setup Frontend
  + Setup React Environment - Ankit Misra
  + Create Employee Main Page Design - Uzair Ahmed
  + Create Home Page Design - Ankit Misra
  + Patient/Employee Database Page - Ankit Misra
* Setup Backend
  + Setup Flask Environment - Bill Wang
  + Create testing APIs - Karthik Gurram
  + Setup Firebase Environment - Sean Samu
* Collect Sprint 1 Documents
  + JIRA Setup - Uzair Ahmed
  + Create Sprint Backlog - Uzair Ahmed
  + Stakeholder Registration – Anmol Sharma
  + Project Management Plan – Anmol Sharma
  + Create Project Charter – Anmol Sharma
  + Create burndown chart – Anmol Sharma
  + Manage Meeting Notes – Anmol Sharma

### Sprint 2

* Create and Connect Frontend Pages
  + Make Employee Page - Uzair Ahmed
  + Make Home Page - Ankit Misra
  + Make Timetable Components - Hasyen Patel
  + Combine Frontend Components into Home page - Hasyen Patel
  + Make Patient Record Form - Hasyen Patel
  + Make new Employee Form/Modal - Ankit Misra
  + Make New Appointment form/Modal - Hasyen Patel
  + Make new Patient Form/Modal - Ankit Misra
* Create and Connect Backend APIs
  + Create Data Object for Employee - Bill Wang
  + Create Data Object for Patient - Bill Wang
  + Create Data Object for Appointment - Karthik Gurram
  + CRUD APIs for Patient Object - Sean Samu
  + CRUD APIs for Employee Object - Sean Samu
  + CRUD APIs for Appointment Object - Akash Patel
  + REST APIs for Patients – Flask Side - Karthik Gurram
  + REST APIs for Employees – Flask Side - Bill Wang
  + REST APIs for Appointments – Flask Side - Karthik Gurram
* Collect Documents for Sprint 2
  + Burn down chart – Anmol Sharma
  + Create Sprint Backlog - Uzair Ahmed
  + Create presentation – Anmol Sharma
  + Manage Meeting Notes – Anmol Sharma

### Sprint 3

* Create and Connect Frontend Pages
  + Combine Frontend Components into Home Page- Hasyen Patel
* Create and Connect Backend APIs
  + CRUD APIs for Patient Object – Part 2 - Sean Seam
  + CRUD APIs for Employee Object – Part 2 - Akash Patel
  + CRUD APIs for Appointment Object – Part 2 - Akash Patel
  + REST APIs for Employees – Flask Side - Karthik Gurram
  + REST APIs for Patients – Flask Side - Karthik Gurram
  + REST APIs for Appointments – Flask Side - Karthik Gurram
* Improve Frontend Pages
  + Complete Nonfunctional UI for Employee Page - Hasyen Patel
  + Complete Nonfunctional UI for Patient Page - Hasyen Patel
* Add Frontend Functionality
  + REST APIs for Employees – React Side - Ankit Misra
  + REST APIs for Appointments – React Side - Ankit Misra
  + REST APIs for Patients – React Side - Hasyen Patel
* QA Testing
  + API Testing in Postman – Rest Functions - Akash Patel
  + API Testing Postman – Firebase functions - Sean Seam
  + Frontend UI Testing – Nonfunctional UI - Ankit Misra
* Collect Documents for Sprint 3
  + Create Sprint Backlog - Uzair Ahmed
  + Manage Meeting Notes and Start Documentation – Anmol Sharma
  + Gantt Chart – Anmol Sharma
  + Burndown Chart – Anmol Sharma

### Sprint 4

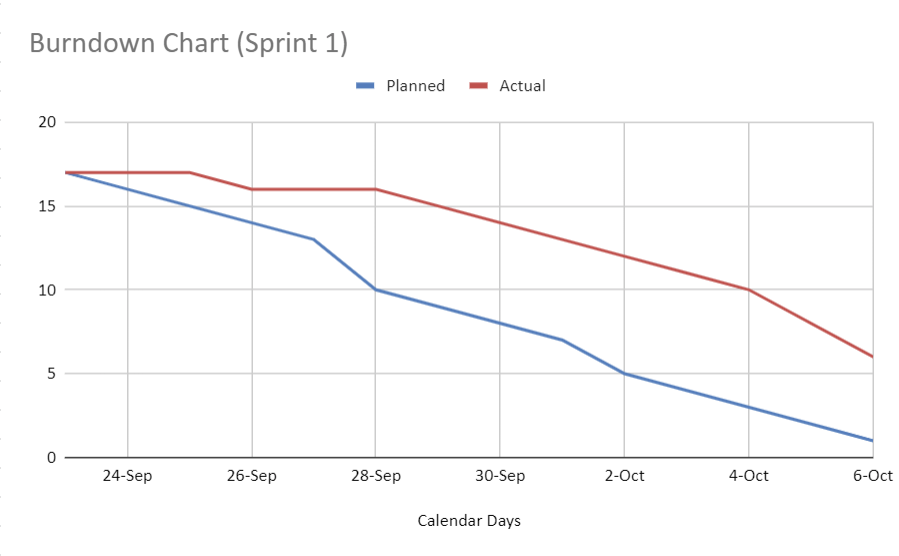
* Add Frontend Functionality
  + Investigate calendar data format - Karthik Gurram
  + Create Data view for patients - Sean Seam
  + Create Data view for employees - Hasyen Patel
  + Improve navigation throughout frontend - Hasyen Patel
  + Clean up CSS - Hasyen Patel
  + Map Patient create form to REST API - Ankit Misra
  + Map Employee create form to REST API - Ankit Misra
  + Map Appointment create form to REST API - Ankit Misra
  + Query Patient for patient records in - Karthik Gurram
  + Login Page - Uzair Ahmed
* Deployment
  + Deploy Server on Heroku instance - Uzair Ahmed
* API Testing
  + API Testing in Postman - Akash Patel
  + API Testing in Postman – Firebase Functions - Sean Seam
  + Frontend UI Testing – Nonfunctional UI - Rubayet Ahmed
  + Basic QA Testing - Rubayet Ahmed
* Collect Documents for Sprint 4
  + Create Sprint Backlog - Uzair Ahmed
  + Create Burn down – Anmol Sharma
  + Create Presentation – Anmol Sharma

### Sprint 5

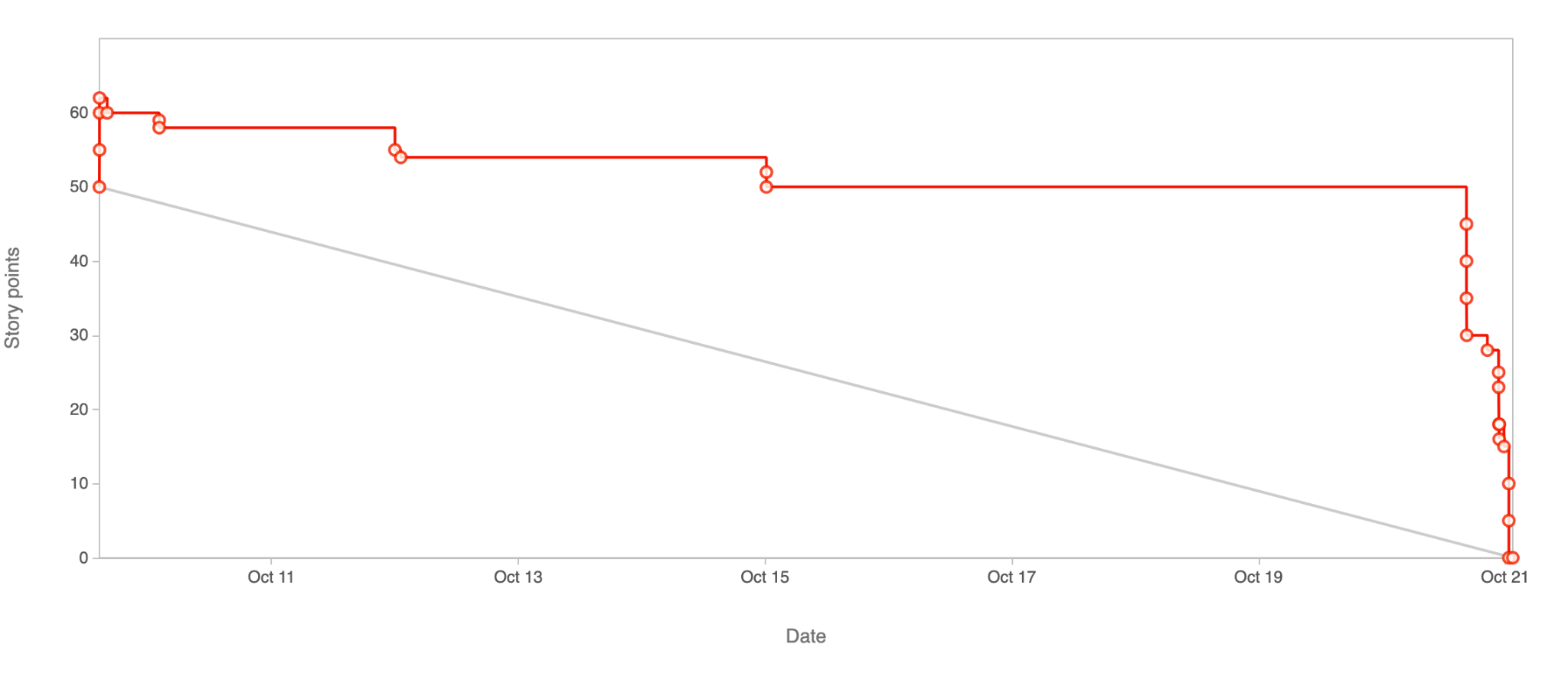
* UI Overhaul
  + Patient Page CSS - Uzair Ahmed
  + Employee Page CSS - Uzair Ahmed
  + Appointment Page CSS - Uzair Ahmed
  + Header Footer CSS - Uzair Ahmed
  + Home Page CSS - Uzair Ahmed
* Add Frontend Functionality
  + Patient Medical Info Page + CSS - Uzair Ahmed
  + Appointments Page Data Mapping - Uzair Ahmed
* Deployment
  + Deploy Frontend App on Netlify Instance - Uzair Ahmed
* End-End Testing
  + Appointments End-End Testing – Akash Patel
  + Employee End-End Testing - Bill Wang
  + Patient End-End Testing – Sean Seam
  + Navigation End-End Testing – Ankit Misra
* Collect Documents for Sprint 5
  + Create Sprint Backlog - Uzair Ahmed
  + Create Report – Anmol Sharma
  + Create Presentation - Anmol Sharma
  + Manage Meeting Notes – Anmol Sharma

## 2.6 - Sprint Burndown Charts

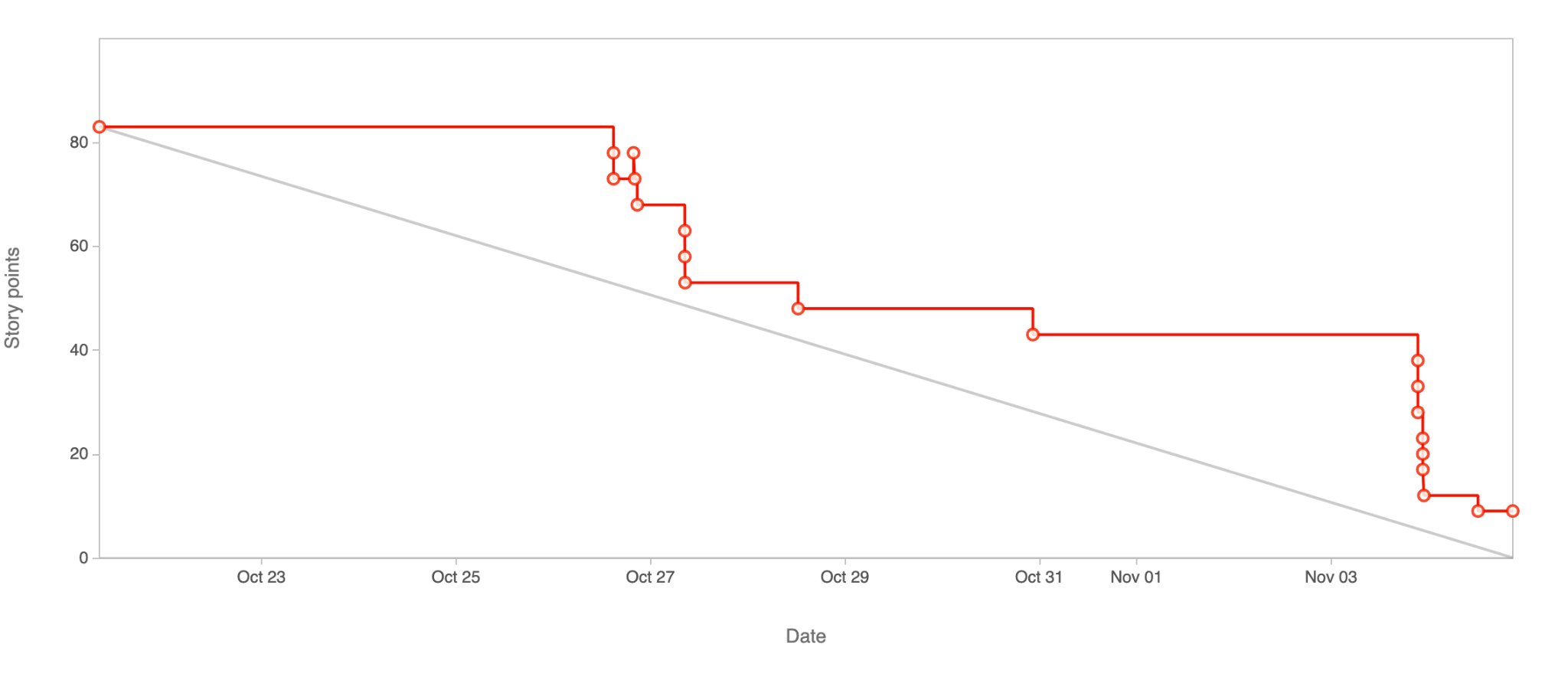
### Sprint 1



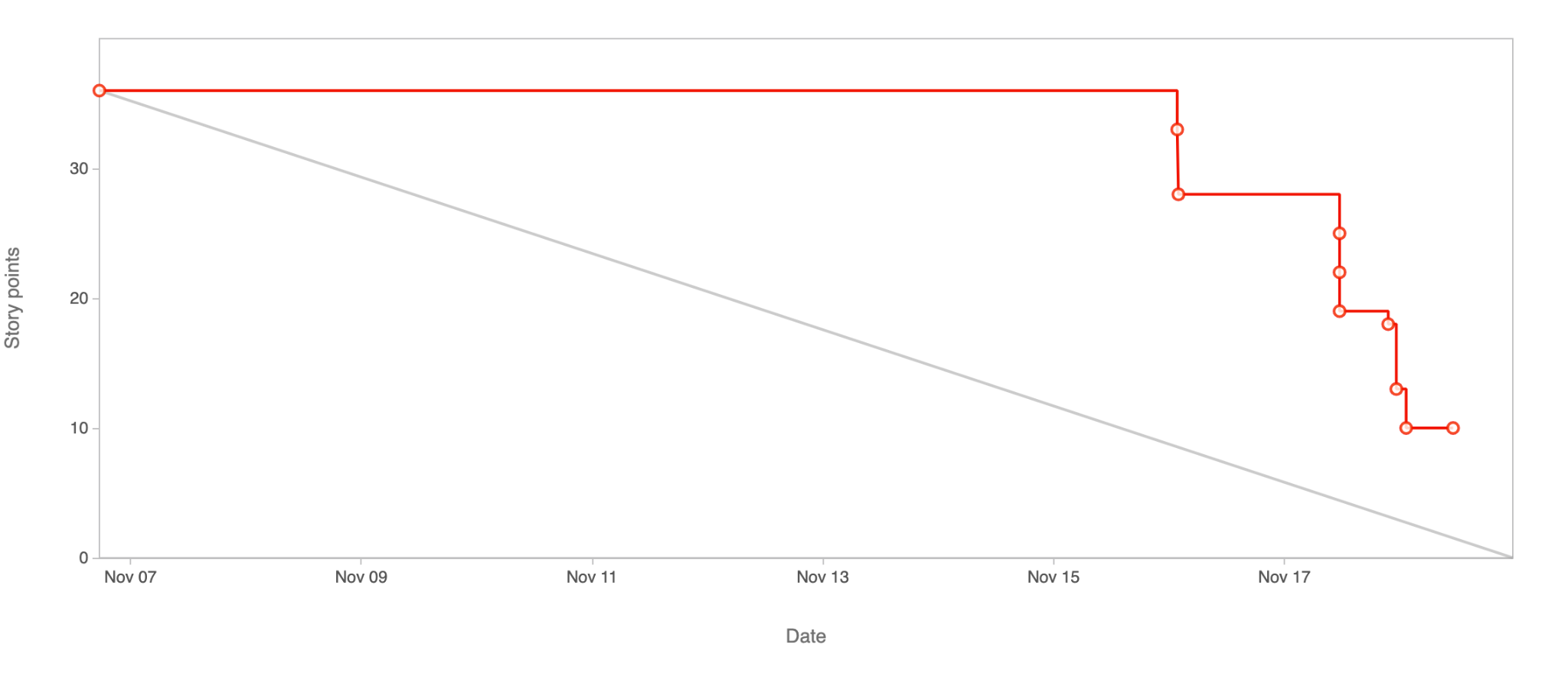
### Sprint 2



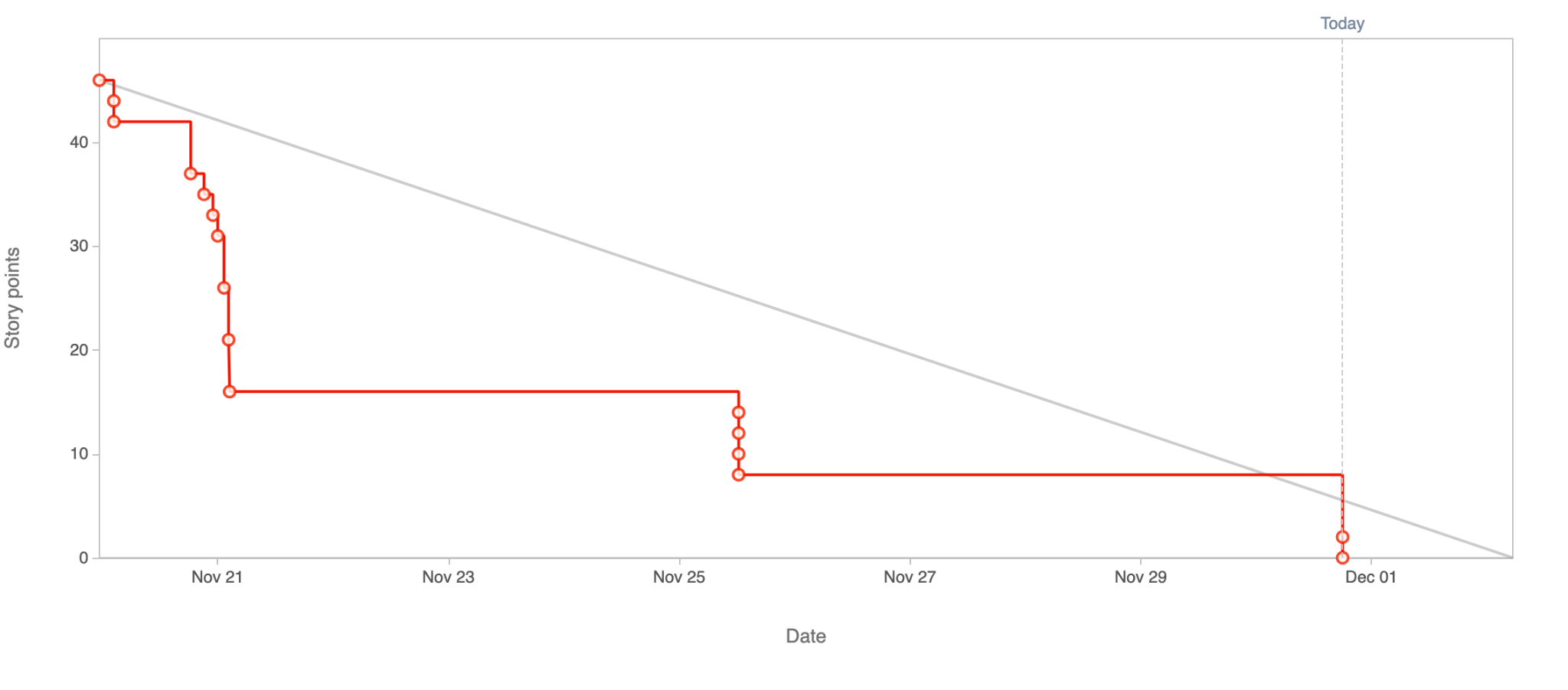
### Sprint 3



### Sprint 4



### Sprint 5

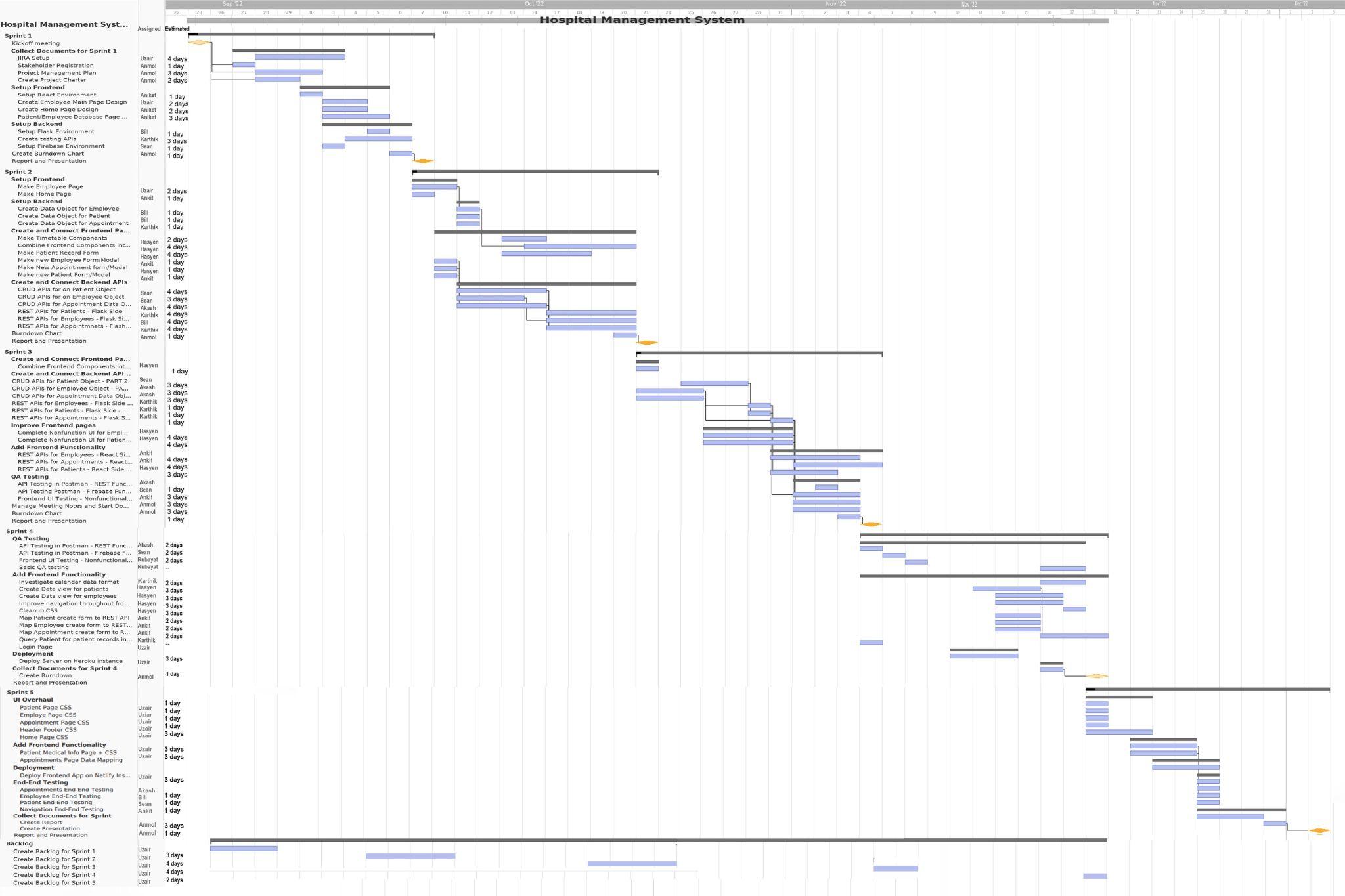


## 2.7 - Velocity report

## 

The team was able to produce on a consistent and committed basis with respect to story points by using the velocity planning approach. The velocity for Sprint 1 was 12, the velocity for Sprint 2 was 59, the velocity for Sprint 3 was 74, the velocity for Sprint 4 was 26, and the velocity for Sprint 5 was 45 story points.

## 2.8 - Gantt Chart



## 2.9 - Meeting Minutes - Sprint Review

### Sprint 4 Review

**Date:** 11/16/2022

**Time:** 10:30 PM - 10:45 PM (15 Mins)

**Meeting Objective:** Review of the last sprint; discussions regarding what work is completed, in progress, or left to do for the upcoming sprint

**Agenda:**

* Go over general sprint goals
* Review Sprint Backlog
* Go over all completed tickets
* Go over In Progress tickets: can it be finished before the sprint ends?
* Go over To-Do tickets: does it need to be moved to the next sprint or does it need a new resource estimate?

| Tickets Completed | Tickets In Progress | Tickets To Do |
| --- | --- | --- |
| **Backend**  Testing   * Appointments API * Patients API * Employee API * Login/Signup API   Deployments   * Backend was deployed   **Frontend**  Functionality   * Patient Form was mapped to API * Appointments Form was mapped to API * Employee Form was mapped to API * Navigation Improvements * Login Page * Calendar Data was investigated | **Frontend**   * Data View for Employee   *Will be completed before sprint 5*   * Data View for Patient   *Will be completed before sprint 5* | **Frontend**   * Unify CSS Styling   *This ticket requires many more resources to complete. Therefore, this ticket will be broken down into multiple tickets, and be carried over to the next sprint.* |

### 

### Sprint 5 Review

**Date:** 11/30/2022

**Time:** 10:30 PM - 10:45 PM (15 Mins)

**Meeting Objective:** Review of the last sprint; discussions regarding what work is completed, in progress, or left to do for the upcoming sprint

**Agenda:**

* Go over general sprint goals
* Review Sprint Backlog
* Go over all completed tickets
* Go over in progress tickets, can it be finished before the sprint ends?
* Go over to do tickets, does it need to be moved to the next sprint or does it need a new resource estimate?

| Tickets Completed | Tickets In Progress | Tickets To Do |
| --- | --- | --- |
| **Frontend**  Unify CSS for   * Appointments * Patients * Employees * Login Signup * Homepage   Deployments   * Frontend was deployed   Functionality   * Remapped APIs * Medical Info Page was mapped   **End-End**  QA Testing   * App was thoroughly tested | *No development tickets in progress, Sprint is complete.* | *No development tickets to do, Sprint is complete* |

## 

## 2.10 - Meeting Minutes - Sprint Retrospective

### Sprint 4 Retrospective

**Date:** 11/16/2022

**Time:** 10:45 PM - 11:00 PM (15 Mins)

**Meeting Objective:** Review of the last sprint; discussions regarding sprint progress and if goals were met; what went well or not well in the sprint; next steps

**Agenda:**

* Go over what went well in the last sprint
* Go over what did not go well in the last sprint
* Go over ideas to improve for next sprint

| What went well? | What did not go well? | What can we do next? |
| --- | --- | --- |
| * We were able to effectively cooperate and strengthen our backend APIs because of the added QA testing done. * When linking backend and frontend APIs, communication was key, and was done effectively in this sprint to integrate the backend APIs with the frontend. | * As shown in the burndown chart, many of the tasks were completed very late because of blockers from other tickets. * The time commitment of the CSS ticket was equal to an Epic, and requires an entire sprint of its own. | * The scrum meetings should effectively go over ALL blockers and work to have them solved ASAP. * The time commitment for large tickets should be estimated more effectively. |

### 

### Sprint 5 Retrospective

**Date:** 11/30/2022

**Time:** 10:45 PM - 11:00 PM (15 Mins)

**Meeting Objective:** Review of the last sprint; discussions regarding sprint progress and if goals were met; what went well or not well in the sprint; next steps

**Agenda:**

* Go over what went well in the last sprint
* Go over what did not go well in the last sprint
* Go over ideas to improve for next sprint

| What went well? | What did not go well? | What can we do next? |
| --- | --- | --- |
| * Progress was timed well; Burndown chart was on a steep decline near the \*start\* of the sprint instead of the end | * Task allocation was not evenly distributed among members.   + Reasoning: all pages had to follow the same set of unified design principles | * Increase collaboration and distribute workload evenly; multiple people should work together on parts originally reserved for one person, balancing out their workload |

## 

## 2.11 Source Code of Final Project

### Live Implementation

Both apps are currently live at the following links:

* Frontend
  + <https://cps-714-g7.netlify.app/>
  + Running on Netlify
* Backend
  + <https://cps-714-hospital-management.herokuapp.com/>
  + Running on Heroku

If you would like to run a local version of the apps, follow the instructions below:

### 

### Backend

1. Ensure that Python 3 is installed (<https://www.python.org/downloads/>)
2. Unzip `backend.zip`
3. In a separate terminal:
   1. Navigate to `[path to folder]`
   2. Run `python -m venv env`
   3. Activate `venv`
      1. On Unix or MacOS `source env/bin/activate`
      2. On Windows `env\Scripts\activate`
   4. Navigate to `./hospital\_management`
   5. Run `python -m pip install -r requirements.txt`
   6. Run `python app.py`
4. In a few seconds, your flask app should start, running at <http://127.0.0.1:5000>

### Frontend

1. Ensure that NodeJS 18 is installed (<https://nodejs.org/en/download/>)
2. Unzip`frontend.zip`
3. In a separate terminal:
   1. Navigate to `[path to folder]/hospital-management`
   2. Run `npm install`
   3. After the install, run `npm start`
4. In a few seconds, your react app should start, and your web browser should open with the React App, running at <http://localhost:3000>
5. By default, the API endpoint for the app is set to the live Heroku server. To test local end-end functionality with a local server:
   1. Navigate to `[path to folder]/hospital-management/src/api.js`
   2. Comment Line 4
   3. Uncomment Line 5

# Part 3 - Technical Description and Documentation

*This section will give a technical overview of the product. The basic functionality of the system will be introduced, as well as its interactions with other systems. The section will also describe system stakeholders and what functionality is available for each user type. At last, the constraints and assumptions for the system will be presented.*

## 3.0 - Product Perspective

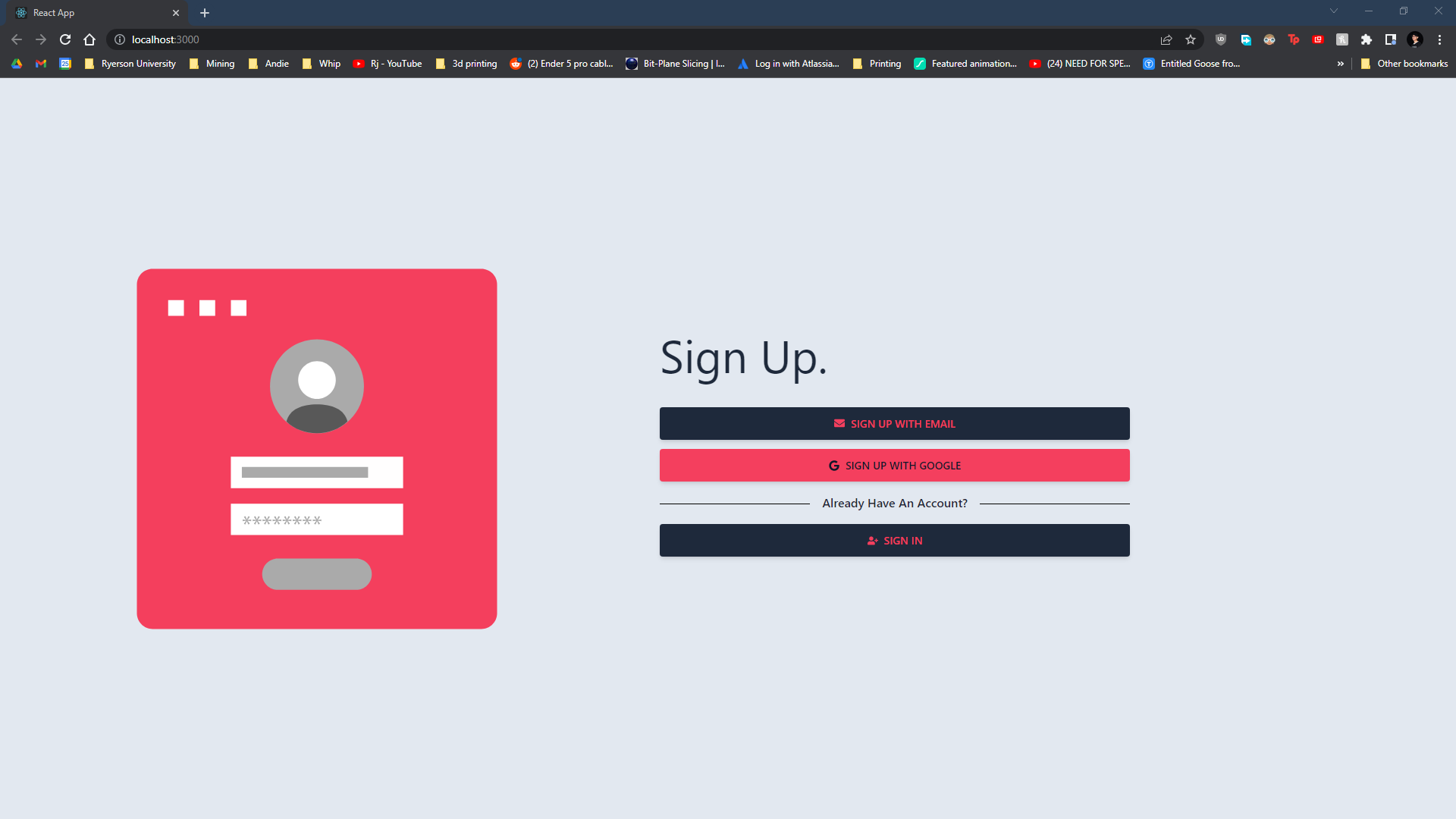
The system consists of one web portal that acts as a user interface for managing information about the hospital employees, patients and appointments and the system as a whole. The web portal can also be used by other users to view any of the information about the employees, patients or appointments.

## 

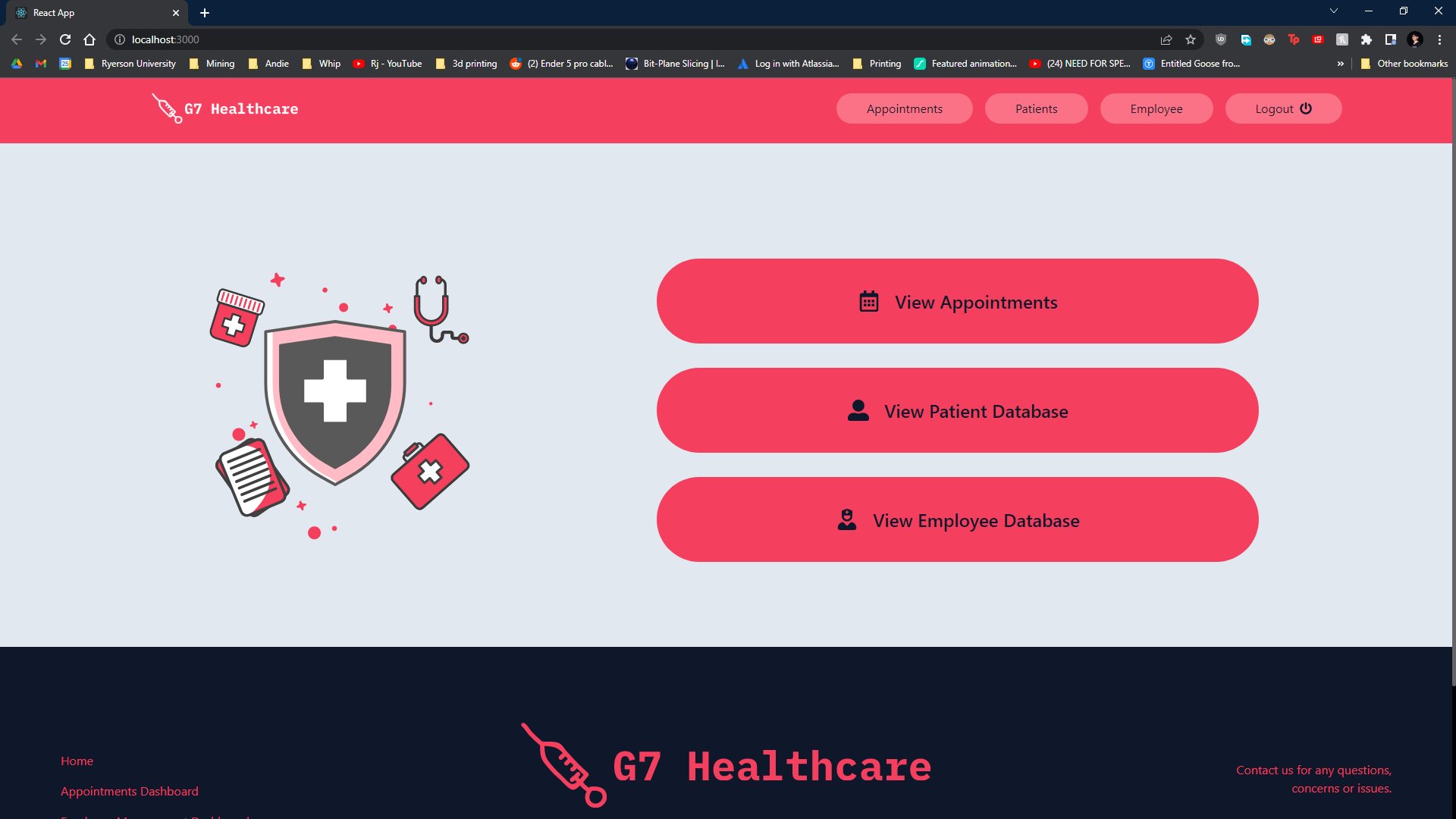
As the product is data-centric in nature, the data will be stored in a database, where data can be accessed by end users through the web portal. The web portal will communicate with the database in order to add and/or modify the data within the database. The web portal will also read the data from the database and display the information for the user to read in a clear and concise format.

## 3.1 - Product Functions

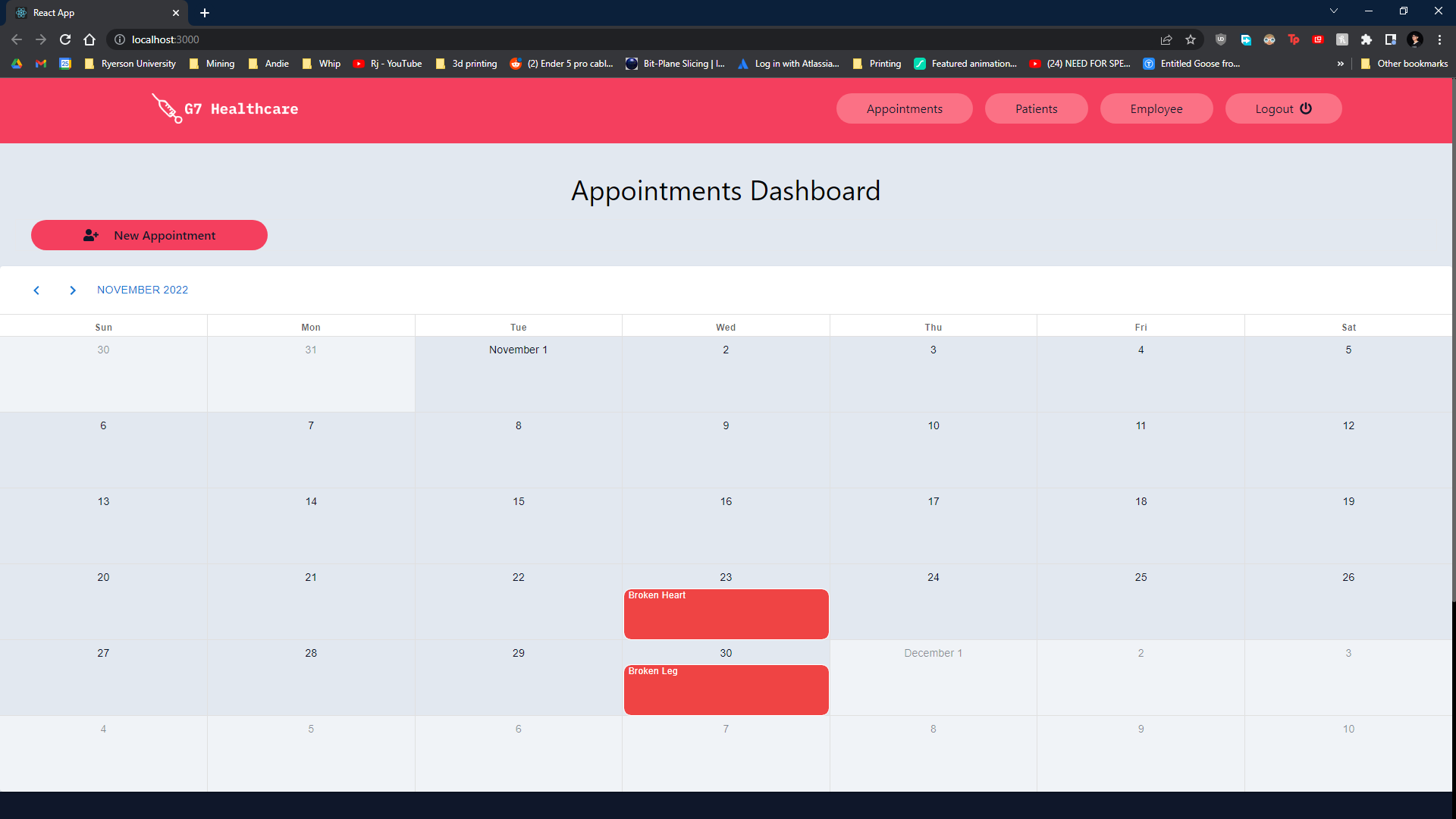
On load, the application first presents the login page, where existing users can login using their email and password, or log in/sign up using Google authentication. There is also an option to sign up using email.



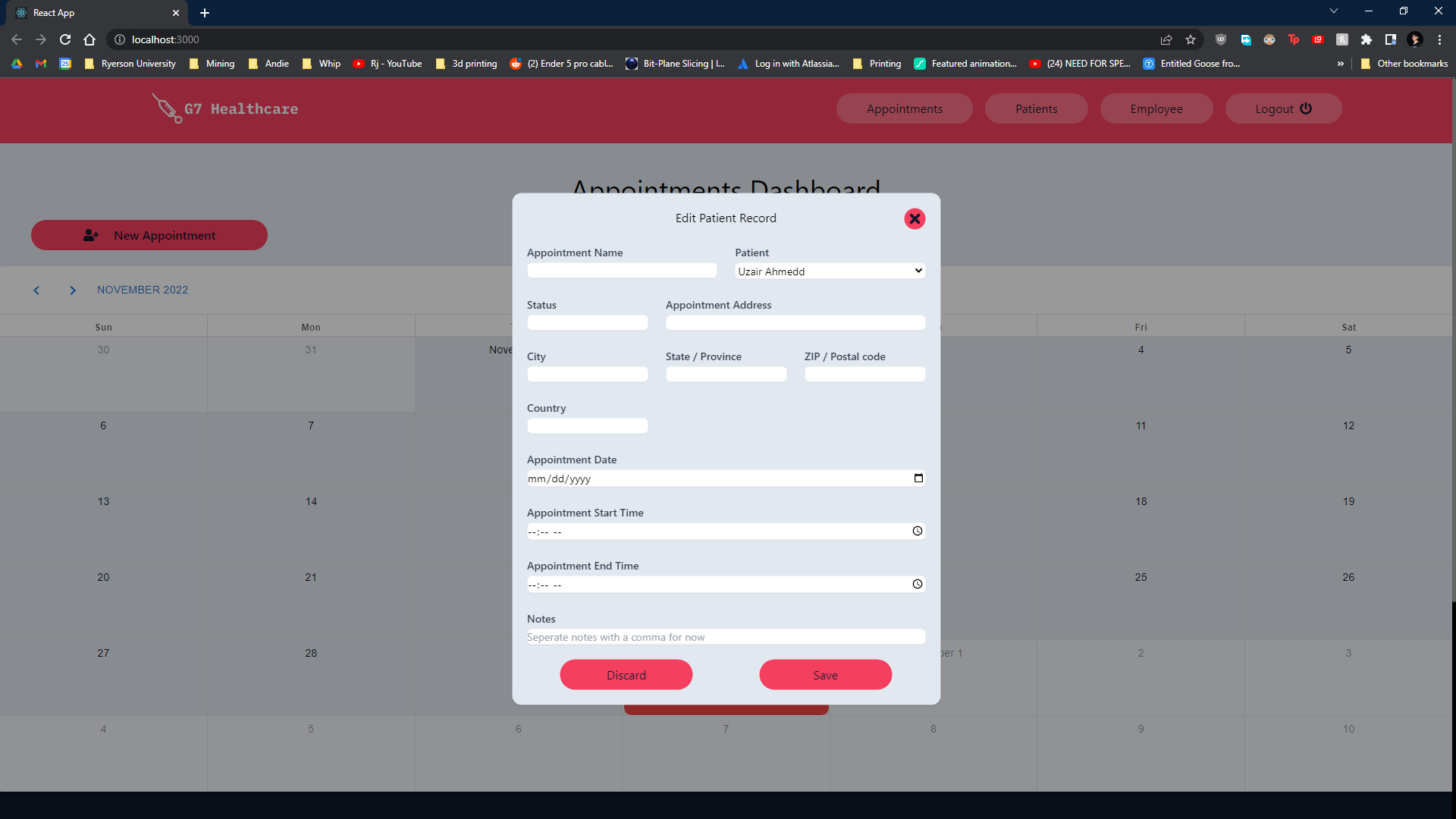
After logging in, the user will be brought to the home page where they can access the appointments calendar, patient lists, and employee lists.



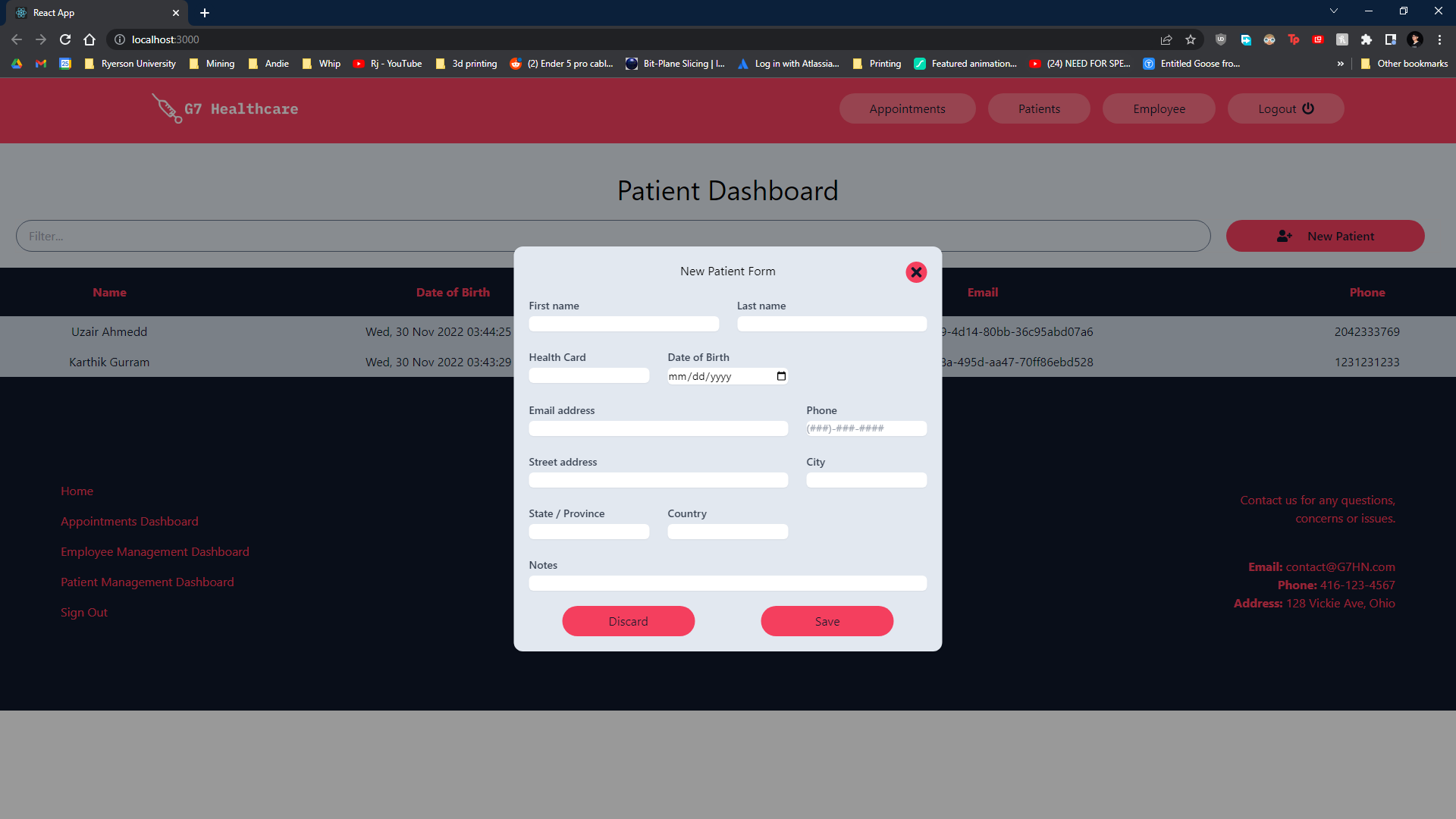
On the Appointments page, a calendar with all patient appointments is displayed. If a user clicks on an appointment, they will be shown the exact time and any notes the doctors have left. Users will be able to move through months and also add new appointments.



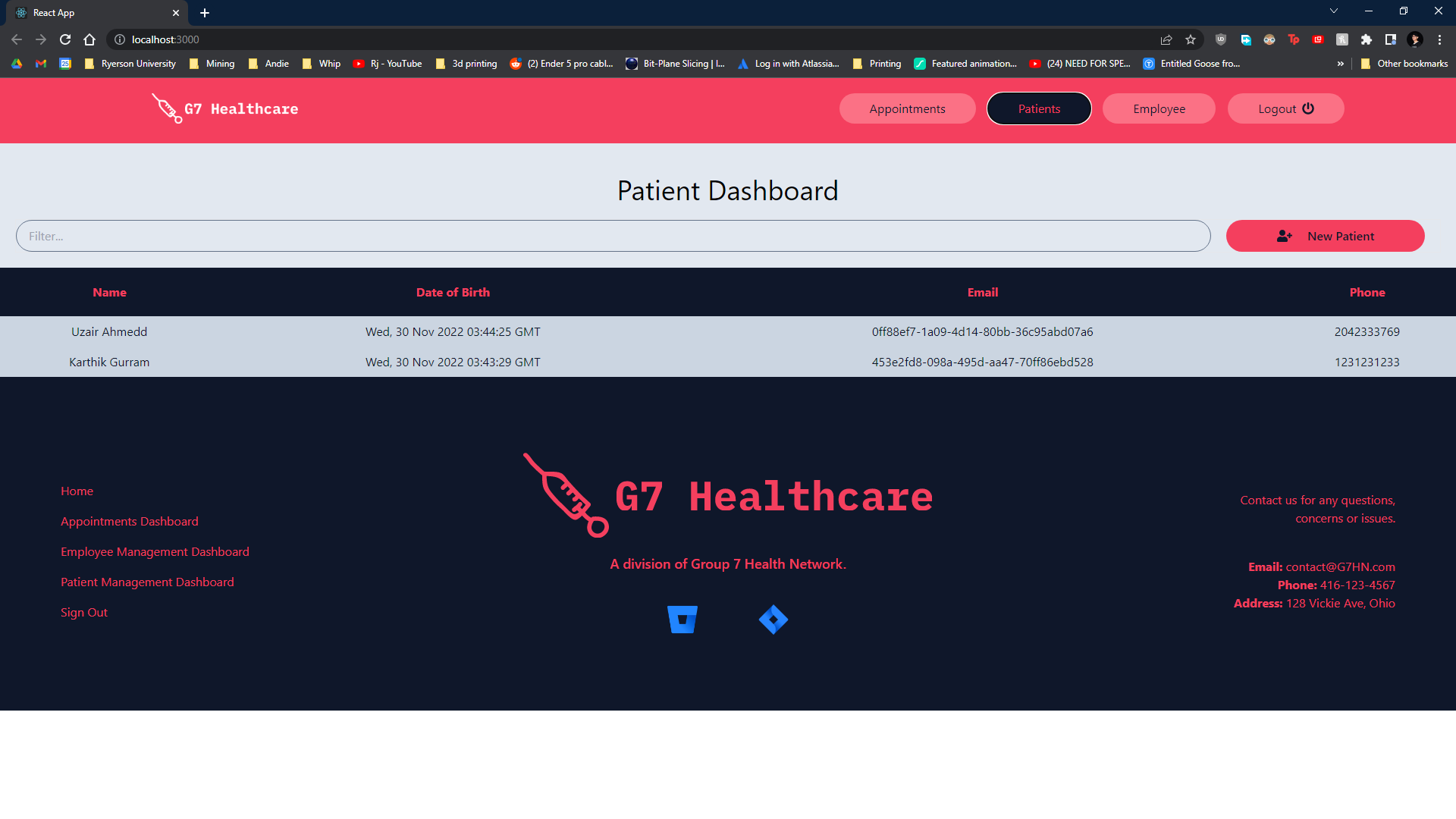
If a user wants to make a new appointment, they will be shown a form that must be fully filled out with their patient information, appointment time and chosen doctor/physician. They can then choose to save the appointment, adding it to the calendar, or discard it.



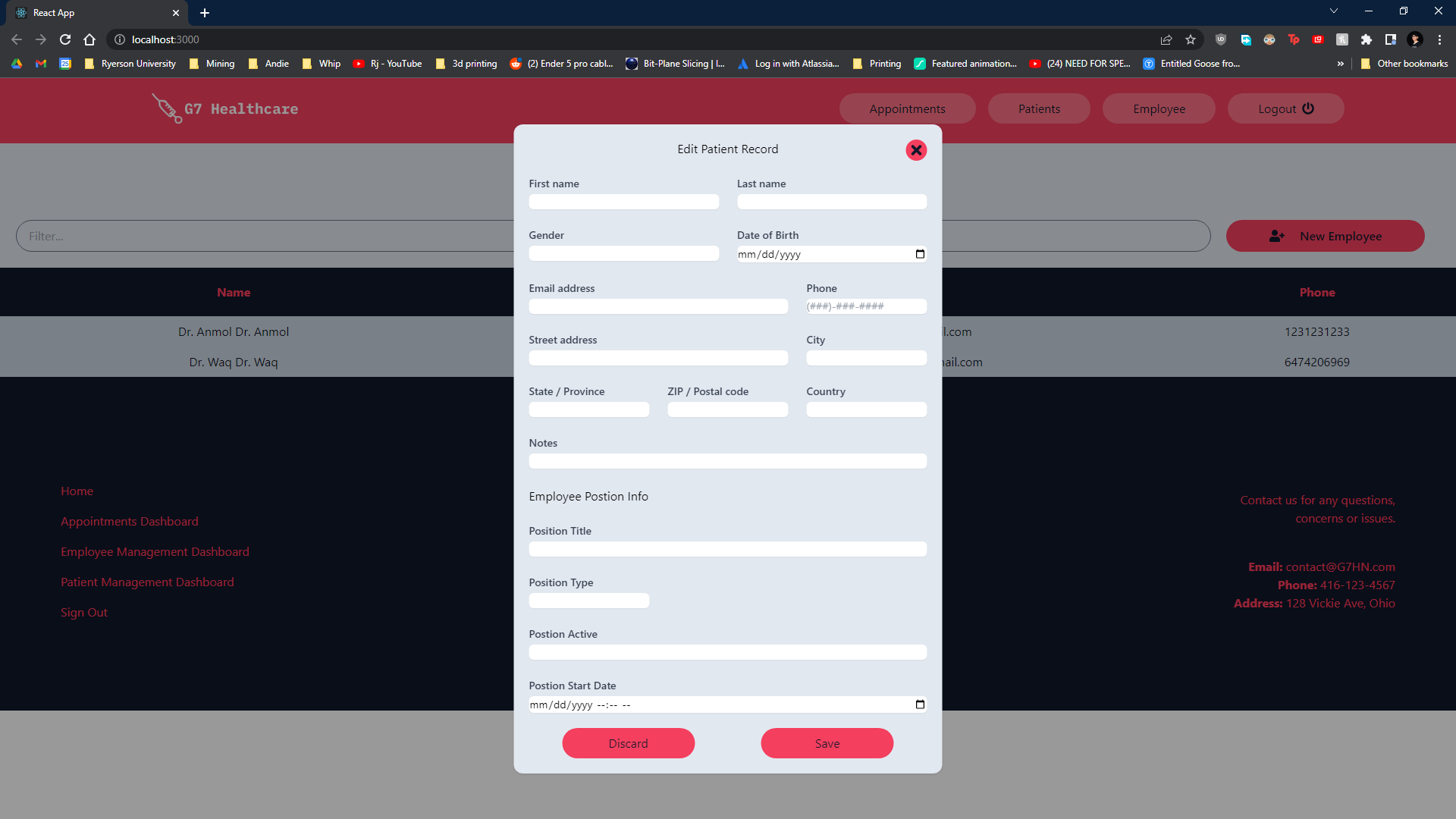
If the hospital staff chooses to create a new patient, they will be presented with an intake form that they must fill out with all of the patient information. They can choose to save the form once they are done or discard the form.



On successful creation, the new patient will appear on the patient dashboard, which can be clicked to view in further detail or to be edited. If the user decides to click on the patient, they can edit any of the patient data and choose to save or discard any updates.



To add a new employee, hospital staff must fill out the form with all the employee’s information. Unlike patients, staff cannot edit employee information after the form is submitted.



On the Employee page, hospital staff will be presented with the employee dashboard which lists all of the registered employees in the organization. From this page, they can choose to add a new employee by clicking on the new employee button similar to how the new patient is created.

